⊗EPA

2DFATMIC

Two-Dimensional Subsurface Flow, Fate and Transport of Microbes and Chemicals Model

User's Manual
Version 1.0

2DFATMIC: User's Manual of a Two-Dimensional Subsurface Flow, Fate and Transport of Microbes and Chemicals Model
Version 1.0

by

Gour-Tsyh (George) Yeh and Jing-Ru (Ruth) Cheng
Department of Civil and Environmental Engineering
The Pennsylvania State University
University Park, PA 16802

and

Thomas E. Short
U. S. Environmental Protection Agency
Robert S. Kerr Environmental Research Center
Subsurface Protection and Remediation Division
Ada, Oklahoma 74820

Project Officer
Thomas E. Short
U. S. Environmental Protection Agency
Robert S. Kerr Environmental Research Center
Subsurface Protection and Remediation Division
Ada, Oklahoma 74820

National Risk Management Research Laboratory
Office of Research and Development
U.S. Environmental Protection Agency
Cincinnati, OH 45268

DISCLAIMER

The U. S. Environmental Protection Agency through its Office of Research and Development partially funded and collaborated in the research described here under assistance agreement number CR-818322 to The Pennsylvania State University. It has been subjected to the Agency's peer and administrative review and has been approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

When available, the software described in this document is supplied on "as-is" basis without guarantee or warranty of any kind, express or implied. Neither the United States Government (United States Environmental Protection Agency, Robert S. Kerr Environmental Research Center), The Pennsylvania State University, nor any of the authors accept any liability resulting from use of this software.

FOREWORD

The U.S. Environmental Protection Agency is charged by Congress with protecting the Nation's land, air, and water resources. Under a mandate of national environmental laws, the Agency strives to formulate and implement actions leading to a compatible balance between human activities and the ability of natural systems to support and nurture life. To meet these mandates, EPA's research program is providing data and technical support for solving environmental problems today and building a science knowledge base necessary to manage our ecological resources wisely, understand how pollutants affect our health, and prevent or reduce environmental risks in the future.

The National Risk Management Research Laboratory is the Agency's center for investigation of technological and management approaches for reducing risks from threats to human health and the environment. The focus of the Laboratory's research program is on methods for the prevention and control of pollution to air, land, water, and subsurface resources; protection of water quality in public water systems; remediation of contaminated sites and ground water, and prevention and control of indoor air pollution. The goal of this research effort is to catalyze development and implementation of innovative, cost-effective environmental technologies; develop scientific and engineering information needed by EPA to support regulatory and policy decisions; and provide technical support and information transfer to ensure effective implementation of environmental regulations and strategies.

Bioremediation is unique among remediation technologies in that it degrades or transforms contaminants through the use, possibly with manipulative enhancement, of indigenous microorganisms. Bioremediation can be used in many ways - degradation on concentrated organic contaminants near their sources, as a secondary remediation strategy following physical or chemical treatment methods, for sequestration of metals through microbially mediated transformation processes, and for remediating large plumes of dilute contaminants that are broadly dispersed in the environment. Thus, bioremediation has the potential to be one of the most cost-effective technologies for dealing with environmental remediation problems. Yet, realistically quantitative predictions and assessments of bioremediation technologies appear lacking. In order to meet the objectives of having a realistic tool for predicting and assessing if a bioremediation technology can be successfully implemented, the 2DFATMIC model has been developed. This numerical model simulates 1) the fate and transport of multiple microbes, electron acceptors, substrates, and nutrients and density-dependent fluid flow in saturated-unsaturated subsurface media under either steady-state or transient conditions; 2) multiple distributed and point sources/sinks as well as boundary sources; and, 3) processes which degrade and transform contaminants, cause the growth and death of microbes, and control the fluid flow.

Clinton W. Hall, Director
Subsurface Protection and Remediation Division
National Risk Management Research Laboratory

iii

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This document is the user's manual of 2DFATMIC, a 2-Dimensional Subsurface Flow, FAte and Transport of MIcrobes and Chemicals Model using a Lagrangian-Eulerian adapted zooming and peak capturing (LEZOOMPC) algorithm. The 2-dimensional model can completely eliminate peak clipping, spurious oscillation, and numerical diffusion, i.e., solve the advective transport problems exactly, within any prescribed error tolerance, using very large mesh Courant numbers. The size of the mesh Courant number is limited only by the accuracy requirement of the Eulerian step. Since this model also includes diffusion zooming in solving diffusion elemental matrix, the accuracy is improved by specifying the number of local subelements in every global element. In other words, the more subelements zoomed in the diffusion step, the more accuracy at the Eulerian step. To sum up, the larger the time-step size is used, the better the solution can be obtained with respect to advection transport; the time-step sizes are only limited by the accuracy requirement with respect to diffusion/dispersion transport and chemical reaction terms. However, the limitation of time-step size imposed by diffusion/dispersion transport is normally not a very severe restriction.

The model, 2DFATMIC, is designed to obtain the density-dependent fluid velocity field, and to solve the advective-dispersive transport equation coupled with biodegradation and microbial biomass production. The water flow through saturated-unsaturated media and the fate and transport of seven components (one substrate, two electron acceptors, one trace element, and three microbial populations) are modeled. For each specific application, 68 maximal control-integers must be assigned using PARAMETER statements in the MAIN program. In addition, if a user uses different analytical forms of boundary conditions, source/sink strength value functions, and soil property functions from those used in this program, he is instructed to modify subroutines ESSFCT, WSSFCT, VBVFCT, DBVFCT, NBVFCT, CBVFCT, and SPFUNC, respectively. The input data to the program include the control indices, properties of the media either in tabular or analytical

form, the geometry in the form of elements and nodes, initial conditions and boundary conditions for flow and transport, and microbe-chemical interaction constants. Principal output includes the spatial distribution of pressure head, total head, moisture content, Darcy velocity component, concentrations, and material fluxes at any desired time step. Fluxes through various types of boundaries are shown in the mass balance table. In addition, diagnostic variables, such as the number of non-convergent nodes and residuals, may be printed if desired for debugging purposes.

1. INTRODUCTION

2DFATMIC (A <u>TWO</u>-<u>D</u>imensional Subsurface <u>F</u>low <u>FA</u>te and <u>T</u>ransport of <u>MI</u>crobes and Chemicals Model) can be used to investigate saturated-unsaturated flow alone, contaminant transport alone, combined flow and transport, or the fate and transport of microbes and chemicals in groundwater environment. For the flow module, the Galerkin finite element method is used to discretize the Richards' equation which governs the flow in variably saturated media. For the transport module, the hybrid Lagrangian-Eulerian approach with an adapted zooming and peak capturing algorithm is used to discretize the transport equation. This approach can completely eliminate spurious oscillation, numerical dispersion, and peak clipping due to advection transport. Large time-step sizes as well as large spatial-grid sizes can be used and still yield accurate simulations. The only limitation on the size of time steps is the requirement of accuracy with respect to dispersion transport which does not pose severe restrictions.

The purpose of this manual is to provide guidance to users of the computer code for their specific application. Section 2.1 lists the governing equations, initial conditions, and boundary conditions for which the 2DFATMIC is designed to solve. Section 2.2 describes the numerical procedure used to simulate the governing equations. Section 2.3 contains the description of all subroutines in 2DFATMIC. Since occasions may arise that requires the user to modify the code, this section should help the user to understand the code structure so the user can make necessary adjustments for individual purposes. Section 3.1 contains the control-parameter specification. For each application, the user needs to assign 68 maximal control-integers in the MAIN program. Section 3.2 describes the required modification of the code so that one might use a different analytical form of soil property function from the ones used in this report. Section 3.3 describes files required for the execution of 2DFATMIC. Appendix A contains the data input guide that is essential for any specific application.

The user may choose whatever consistent units he wants to. Units of mass (M), length (L), and time (T) are indicated in the input description.

The special features of 2DFATMIC are its flexibility and versatility in modeling as wide a range of problems as possible. In terms of media, this model can handle heterogeneous and anisotropic media consisting of as many geologic formations as desired. In terms of source/sink, this model includes both distributed and point sources/sinks that are spatially and temporally dependent. In terms of initial conditions, the model can take input initial values or obtain initial values by simulating a steady state version of the system under consideration. In terms of boundary conditions, the model can handle: 1) the prescribed transient concentration over Dirichlet nodes; 2) time dependent fluxes over Neumann nodes; 3) time dependent total fluxes over Cauchy nodes; and 4) variable boundary conditions of evaporation, infiltration, or seepage on the soil-air interface for the flow module and variable boundary conditions of inflow and outflow for the transport module automatically. In terms of composing the matrix, the model provides two options to treat the mass matrix: consistent and lumping. In terms of linearization, the model gives three options for estimating the nonlinear matrix: 1) exact relaxation, 2) under-relaxation, and 3) over-relaxation. To properly treat the advection terms, the model provides three options: 1) the Galerkin weighting, 2) the upstream weighting, and 3) the Lagrangian approach. To consider the balance between accuracy and efficiency, the model includes two options in the Lagrangian step of solving the advective transport: enabling and disabling adapted zooming scheme. It also includes two options in the Eulerian step of solving the diffusion problems: enable and disable of diffusion zooming. The model can automatically reset the time step size when boundary conditions or sources/sinks have changed abruptly. It checks the mass balance over the entire region for every time step. Finally, the source program can easily be modified if microbial processes different from those used in this model are used.

2. DESCRIPTION OF 2DFATMIC MODEL

2.1 Mathematical Statement of 2DFATMIC

2DFATMIC is designed to solve the following system of governing equations along with initial and boundary conditions, which describe flow and transport through saturated-unsaturated media. The governing equations for flow, which describe the flow of a variable-density fluid, are basically the Richards' equation.

Governing Flow Equation

$$\frac{\rho}{\rho_w}\left(\alpha'\frac{\theta}{n_e} + \beta'\theta + n_e\frac{dS}{dh}\right)\frac{\partial h}{\partial t} = \nabla\cdot[\mathbf{K_s}K_r\cdot(\nabla h + \frac{\rho}{\rho_w}\nabla z)] + \frac{\rho^*}{\rho_w}q\,(\text{or} \; -\frac{\rho}{\rho_w}q) \tag{2.1}$$

in which the saturated hydraulic conductivity $\mathbf{K_s}$ is given by

$$\mathbf{K_s} = \mathbf{K_{sw}}\frac{(\rho/\rho_w)}{(\mu/\mu_w)} \tag{2.2a}$$

where θ is the moisture content, α' is the modified compressiblity of the media (1/L), β' is the modified compressibility of water (1/L), n_e is the effective porosity, h is the referenced pressure head defined as $p/\rho_w g$ in which p is pressure (M/LT^2), t is time (T), $\mathbf{K_s}$ is the saturated hydraulic conductivity tensor (L/T), K_r is the relative hydraulic conductivity or relative permeability, z is the potential head (L), q is the source and/or sink (L^3/T/L), and θ is the moisture content, ρ and μ are the density (M/L^3) and dynamic viscosity (M/LT) at microbial concentrations C_1, C_2, C_3, and chemical concentrations C_s, C_o, C_n, and C_p (M/L^3); $\mathbf{K_{sw}}$, ρ_w and μ_w are the reference saturated hydraulic conductivity tensor, density, and dynamic viscosity, respectively. The values of reference saturated hydraulic conductivity are usually taken as the saturated hydraulic conductivity at zero

microbial and chemical concentrations. The strength of source/sink is the discharge or withdraw flow rate q,

and ρ^* is the density of injected fluid. It should be noted that in this version of 2DFATMIC, α' and β' were

assumed to be zero, i.e., the media were assumed rigid.

The density and dynamic viscosity of fluid are functions of microbial and chemical concentrations and

are assumed to take the following form (See Appendix B in 3DFATMIC):

$$\frac{\rho}{\rho_w} = 1 + \sum_i \left(\frac{1}{\rho_w} - \frac{1}{\rho_i} \right) C_i \; ; \quad i = 1, 2, 3, s, o, n, p \qquad (2.2b)$$

$$\frac{\mu}{\mu_w} = 1 + \beta_s C_s + \beta_o C_o + \beta_n C_n + \beta_p C_p + \beta_1 C_1 + \beta_2 C_2 + \beta_3 C_3 \qquad (2.2c)$$

where C_s and ρ_s are dissolved concentration and intrinsic density of the substrate, respectively (M/L³); C_o and

ρ_o are dissolved concentration and intrinsic density of oxygen (M/L³), respectively; C_n and ρ_n are dissolved

concentration and intrinsic density of nitrate (M/L³), respectively; C_p and ρ_p are dissolved concentration and

intrinsic density of nutrient (M/L³), respectively; C_1 and ρ_1 are dissolved concentration and intrinsic density

of microbe #1 (M/L³), respectively; C_2 and ρ_2 are dissolved concentration and intrinsic density of microbe #2

(M/L³), respectively; C_3 and ρ_3 are dissolved concentration and intrinsic density of microbe #3 (M/L³),

respectively; β_s, β_o, β_n, β_p, β_1, β_2, and β_3 are factors representing the effect of associated species-concentrations

on the viscosity (L²/T). It is assumed that microbe #1 utilizes the substrate under aerobic conditions, microbe

#2 utilizes the substrate under anaerobic conditions, and microbe #3 utilizes the substrate under both aerobic

and anaerobic conditions.

The Darcy velocity is calculated as follows.

$$\mathbf{V} = -\mathbf{K}_s K_r \cdot (\frac{\rho_w}{\rho} \nabla h + \nabla z) \qquad (2.3)$$

4

<u>Initial Conditions for Flow Equation</u>

$$h = h_i(x,z) \qquad \text{in } R \tag{2.4}$$

where R is the region of interest and h_i is the prescribed initial condition which can be obtained by either field measurement or by solving the steady state version of Eq. (2.1).

<u>Boundary Conditions for Flow Equation</u>

Dirichlet Conditions:

$$h = h_d(x_b, z_b, t) \qquad \text{on } B_d \tag{2.5}$$

Neumann Conditions:

$$-\mathbf{n} \cdot \mathbf{K_s} K_r \cdot \frac{\rho_w}{\rho} \nabla h = q_n(x_b, z_b, t) \qquad \text{on } B_n \tag{2.6}$$

Cauchy Conditions:

$$-\mathbf{n} \cdot \mathbf{K_s} K_r \cdot (\frac{\rho_w}{\rho} \nabla h + \nabla z) = q_c(x_b, z_b, t) \qquad \text{on } B_c \tag{2.7}$$

Variable Conditions - During Precipitation Period:

$$h = h_p(x_b, z_b, t) \qquad \textit{iff} \quad -\mathbf{n} \cdot \mathbf{K_s} K_r (\frac{\rho_w}{\rho} \nabla h + \nabla z) \geq q_p \qquad \text{on } B_v \tag{2.8a}$$

or

$$-\mathbf{n} \cdot \mathbf{K_s} K_r \cdot (\frac{\rho_w}{\rho} \nabla h + \nabla z) = q_p(x_b, z_b, t) \quad \text{iff} \quad h \leq h_p \qquad \text{on } B_v \tag{2.8b}$$

Variable Conditions - During Non-precipitation period:

$$h = h_p(x_b, z_b, t) \quad \textit{iff} \quad \mathbf{n} \cdot \mathbf{K_s} \cdot K_r (\frac{\rho_w}{\rho} \nabla h + \nabla z) \geq 0 \qquad \text{on } B_v \tag{2.8c}$$

or

5

$$h = h_m(x_b, z_b, t) \quad \textit{iff} \quad \mathbf{n} \cdot \mathbf{K_s} \cdot K_r \cdot (\frac{\rho_w}{\rho} \nabla h + \nabla z) \leq q_e \qquad \text{on } B_v \qquad (2.8d)$$

or

$$-\mathbf{n} \cdot \mathbf{K_s} K_r \cdot (\frac{\rho_w}{\rho} \nabla h + \nabla z) = q_e(x_b, z_b, t) \quad \textit{iff} \quad h \geq h_m \qquad \text{on } B_v \qquad (2.8e)$$

where (x_b, z_b) is the spatial coordinate on the boundary; $\mathbf{n}$ is an outward unit vector normal to the boundary; h_d, q_n, and q_c are the prescribed Dirichlet functional values, Neumann fluxes, and Cauchy fluxes, respectively; B_d, B_n, and B_c are the Dirichlet, Neumann, and Cauchy boundaries, respectively; B_v is the variable boundary; h_p is the allowed ponding depth and q_p is the throughfall of precipitation on the variable boundary; h_m is the allowed minimum pressure and q_e is the allowed maximum evaporation rate on the variable boundary, which is the potential evaporation. Only one of Eqs. (2.8a) through (2.8e) is used at any point on the variable boundary at any time.

The governing equations for transport are derived based on the continuity of mass and flux laws. The major processes are advection, dispersion/diffusion, adsorption, decay, source/sink, and microbial-chemical interactions.

Governing Equations for Transport

Transport of the substrate, oxygen, nitrate and nutrient in the bulk pore fluid is expressed by advection-dispersion equations that are coupled sink terms that account for biodegradation. The biodegradation is modeled with the modified Monod kinetics. The four nonlinear transport and fate equations are

$$\left(\theta + \rho_b K_{ds}\right)\frac{\partial C_s}{\partial t} + \mathbf{V}\cdot\nabla C_s = \nabla\cdot\theta\mathbf{D}\cdot\nabla C_s - \Lambda_s(\theta+\rho_b K_{ds})C_s + q_{in}C_{sin} + \left(\frac{\rho_w}{\rho}\mathbf{V}\cdot\nabla(\frac{\rho}{\rho_w}) - \frac{\rho^*}{\rho}q_{in}\right)C_s$$

$$-\left[\left(\theta + \rho_b K_{d1}\right)C_1\right]\left\{\frac{\mu_o^{(1)}}{Y_o^{(1)}}\left[\frac{C_s}{K_{so}^{(1)}+C_s}\right]\left[\frac{C_o}{K_o^{(1)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(1)}+C_p}\right]\right\}$$

$$-\left[\left(\theta + \rho_b K_{d2}\right)C_2\right]\left\{\frac{\mu_n^{(2)}}{Y_n^{(2)}}\left[\frac{C_s}{K_{sn}^{(2)}+C_s}\right]\left[\frac{C_n}{K_n^{(2)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(2)}+C_p}\right]\right\} \qquad (2.9)$$

$$-\left[\left(\theta + \rho_b K_{d3}\right)C_3\right]\left\{\begin{array}{c}\dfrac{\mu_o^{(3)}}{Y_o^{(3)}}\left[\dfrac{C_s}{K_{so}^{(3)}+C_s}\right]\left[\dfrac{C_o}{K_o^{(3)}+C_o}\right]\left[\dfrac{C_p}{K_{po}^{(3)}+C_p}\right] \\[2em] +\dfrac{\mu_n^{(3)}}{Y_n^{(3)}}\left[\dfrac{C_s}{K_{sn}^{(3)}+C_s}\right]\left[\dfrac{C_n}{K_n^{(3)}+C_n}\right]\left[\dfrac{C_p}{K_{pn}^{(3)}+C_p}\right]I(C_o)\end{array}\right\}$$

$$\left(\theta + \rho_b K_{do}\right)\frac{\partial C_o}{\partial t} + \mathbf{V}\cdot\nabla C_o = \nabla\cdot\theta\mathbf{D}\cdot\nabla C_o - \Lambda_o(\theta+\rho_b K_{do})C_o + q_{in}C_{oin} + \left(\frac{\rho_w}{\rho}\mathbf{V}\cdot\nabla(\frac{\rho}{\rho_w}) - \frac{\rho^*}{\rho}q_{in}\right)C_o$$

$$-\left[\left(\theta + \rho_b K_{d1}\right)C_1\right]\left\{\gamma_o^{(1)}\mu_o^{(1)}\left[\frac{C_s}{K_{so}^{(1)}+C_s}\right]\left[\frac{C_o}{K_o^{(1)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(1)}+C_p}\right] + \alpha_o^{(1)}\lambda_o^{(1)}\left[\frac{C_o}{\Gamma_o^{(1)}+C_o}\right]\right\} \qquad (2.10)$$

$$-\left[\left(\theta + \rho_b K_{d3}\right)C_3\right]\left\{\gamma_o^{(3)}\mu_o^{(3)}\left[\frac{C_s}{K_{so}^{(3)}+C_s}\right]\left[\frac{C_o}{K_o^{(3)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(3)}+C_p}\right] + \alpha_o^{(3)}\lambda_o^{(3)}\left[\frac{C_o}{\Gamma_o^{(3)}+C_o}\right]\right\}$$

$$\left(\theta + \rho_b K_{dn}\right)\frac{\partial C_n}{\partial t} + \mathbf{V}\cdot\nabla C_n = \nabla\cdot\theta\mathbf{D}\cdot\nabla C_n - \Lambda_n(\theta+\rho_b K_{dn})C_n + q_{in}C_{nin} + \left(\frac{\rho_w}{\rho}\mathbf{V}\cdot\nabla(\frac{\rho}{\rho_w}) - \frac{\rho^*}{\rho}q_{in}\right)C_n$$

$$-\left[\left(\theta + \rho_b K_{d2}\right)C_2\right]\left\{\gamma_n^{(2)}\mu_n^{(2)}\left[\frac{C_s}{K_{sn}^{(2)}+C_s}\right]\left[\frac{C_n}{K_n^{(2)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(2)}+C_p}\right] + \alpha_n^{(2)}\lambda_n^{(2)}\left[\frac{C_n}{\Gamma_n^{(2)}+C_n}\right]\right\} \qquad (2.11)$$

$$-\left[\left(\theta + \rho_b K_{d3}\right)C_3\right]\left\{\gamma_n^{(3)}\mu_n^{(3)}\left[\frac{C_s}{K_{sn}^{(3)}+C_s}\right]\left[\frac{C_n}{K_n^{(3)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(3)}+C_p}\right] + \alpha_n^{(3)}\lambda_n^{(3)}\left[\frac{C_n}{\Gamma_n^{(3)}+C_n}\right]\right\}$$

$$\left(\theta + \rho_b K_{dp}\right)\frac{\partial C_p}{\partial t} + \mathbf{V}\cdot\nabla C_p \ = \ \nabla\cdot\theta\mathbf{D}\cdot\nabla C_p - \Lambda_p(\theta + \rho_b K_{dp})C_p + q_{in}C_{pin} + \left(\frac{\rho_w}{\rho}\mathbf{V}\cdot\nabla(\frac{\rho}{\rho_w}) - \frac{\rho^*}{\rho}q_{in}\right)C_p$$

$$- \left[(\theta + \rho_b K_{d1})C_1\right]\left\{\epsilon_o^{(1)}\frac{\mu_o^{(1)}}{Y_o^{(1)}}\left[\frac{C_s}{K_{so}^{(1)} + C_s}\right]\left[\frac{C_o}{K_o^{(1)} + C_o}\right]\left[\frac{C_p}{K_{po}^{(1)} + C_p}\right]\right\}$$

$$- \left[(\theta + \rho_b K_{d2})C_2\right]\left\{\epsilon_n^{(2)}\frac{\mu_n^{(2)}}{Y_n^{(2)}}\left[\frac{C_s}{K_{sn}^{(2)} + C_s}\right]\left[\frac{C_n}{K_n^{(2)} + C_n}\right]\left[\frac{C_p}{K_{pn}^{(2)} + C_p}\right]\right\} \qquad (2.12)$$

$$- \left[(\theta + \rho_b K_{d3})C_3\right]\left\{\begin{array}{l}\epsilon_o^{(3)}\frac{\mu_o^{(3)}}{Y_o^{(3)}}\left[\frac{C_s}{K_{so}^{(3)} + C_s}\right]\left[\frac{C_o}{K_o^{(3)} + C_o}\right]\left[\frac{C_p}{K_{po}^{(3)} + C_p}\right] \\ + \epsilon_n^{(3)}\frac{\mu_n^{(3)}}{Y_n^{(3)}}\left[\frac{C_s}{K_{sn}^{(3)} + C_s}\right]\left[\frac{C_n}{K_n^{(3)} + C_n}\right]\left[\frac{C_p}{K_{pn}^{(3)} + C_p}\right]I(C_o)\end{array}\right\}$$

The transport equations for the three microbes can be derived based on mass-balance of microbes and are given as

$$\left(\theta + \rho_b K_{d1}\right)\frac{\partial C_1}{\partial t} + \mathbf{V}\cdot\nabla C_1 \ = \ \nabla\cdot\theta\mathbf{D}\cdot\nabla C_1 - \Lambda_1(\theta + \rho_b K_{d1})C_1 + q_{in}C_{1in} + \left(\frac{\rho_w}{\rho}\mathbf{V}\cdot\nabla(\frac{\rho}{\rho_w}) - \frac{\rho^*}{\rho}q_{in}\right)C_1$$

$$+ \left(\theta + \rho_b K_{d1}\right)C_1\left\{\mu_o^{(1)}\left[\frac{C_s}{K_{so}^{(1)} + C_s}\right]\left[\frac{C_o}{K_o^{(1)} + C_o}\right]\left[\frac{C_p}{K_{po}^{(1)} + C_p}\right] - \lambda_o^{(1)}\right\} \qquad (2.13)$$

$$\left(\theta + \rho_b K_{d2}\right)\frac{\partial C_2}{\partial t} + \mathbf{V}\cdot\nabla C_2 \ = \ \nabla\cdot\theta\mathbf{D}\cdot\nabla C_2 - \Lambda_2(\theta + \rho_b K_{d2})C_2 + q_{in}C_{2in} + \left(\frac{\rho_w}{\rho}\mathbf{V}\cdot\nabla(\frac{\rho}{\rho_w}) - \frac{\rho^*}{\rho}q_{in}\right)C_2$$

$$+ \left(\theta + \rho_b K_{d2}\right)C_2\left\{\mu_n^{(2)}\left[\frac{C_s}{K_{sn}^{(2)} + C_s}\right]\left[\frac{C_n}{K_n^{(2)} + C_n}\right]\left[\frac{C_p}{K_{pn}^{(2)} + C_p}\right] - \lambda_n^{(2)}\right\} \qquad (2.14)$$

8

$$\left(\theta + \rho_b K_{d3}\right)\frac{\partial C_3}{\partial t} + \mathbf{V}\cdot\nabla C_3 \;=\; \nabla\cdot\theta\mathbf{D}\cdot\nabla C_3 - \Lambda_3(\theta + \rho_b K_{d3})C_3 + q_{in}C_{3in} + \left(\frac{\rho_w}{\rho}\mathbf{V}\cdot\nabla(\frac{\rho}{\rho_w}) - \frac{\rho^*}{\rho}q_{in}\right)C_3$$

$$+ \left(\theta + \rho_b K_{d3}\right)C_3 \left\{ \begin{array}{l} \mu_o^{(3)}\left[\dfrac{C_s}{K_{so}^{(3)} + C_s}\right]\left[\dfrac{C_o}{K_o^{(3)} + C_o}\right]\left[\dfrac{C_p}{K_{po}^{(3)} + C_p}\right] - \lambda_o^{(3)} \\[20pt] + \mu_n^{(3)}\left[\dfrac{C_s}{K_{sn}^{(3)} + C_s}\right]\left[\dfrac{C_n}{K_n^{(3)} + C_n}\right]\left[\dfrac{C_p}{K_{pn}^{(3)} + C_p}\right]I(C_o) - \lambda_n^{(3)}I(C_o) \end{array} \right\} \qquad (2.15)$$

where θ is the moisture content, ρ_b is the bulk density of the medium (M/L^3), t is time, $\mathbf{V}$ is the discharge (L/T), ∇ is the del operator, $\mathbf{D}$ is the dispersion coefficient tensor (L^2/T), where Λ_s, Λ_o, Λ_p, Λ_n, Λ_1, Λ_2, Λ_3 and K_{ds}, K_{do}, K_{dn}, K_{dp}, K_{d1}, K_{d2}, K_{d3} are the transformation rate constants and distribution coefficient of the dissolved substrate, oxygen, nitrate, nutrient, microbe #1, microbe #2, and microbe #3, Q_{in} is the source rate of water, C_{sin}, C_{oin}, C_{nin}, C_{pin}, C_{1in}, C_{2in}, and C_{3in}, are the concentrations of substrate, oxygen, nitrogen, nutrient, microbe #1, microbe #2 and microbe #3 in the source.

The dispersion coefficient tensor $\mathbf{D}$ in Eq. (2.9) to Eq. (2.15) is given by

$$\theta\mathbf{D} \;=\; a_T|\mathbf{V}|\delta + (a_L - a_T)\mathbf{V}\mathbf{V}/|\mathbf{V}| + \theta a_m \tau\delta \qquad (2.16)$$

where $|\mathbf{V}|$ is the magnitude of $\mathbf{V}$, δ is the Kronecker delta tensor, a_T is lateral dispersivity, a_L is the longitudinal dispersivity, a_m is the molecular diffusion coefficient, and τ is the tortuosity. $I(C_0) = [\,1 + C_0/K_c]^{-1}$ is an inhibition function which is under the assumption that denitrifying enzyme inhibition is reversible and noncompetitive, where K_c is inhibition coefficient (M/L^3). $\mu_o^{(1)}$, $\mu_n^{(2)}$, $\mu_o^{(3)}$ and $\mu_n^{(3)}$ are the maximum specific oxygen-based growth rates for microbe #1, the maximum specific nitrate-based growth rate for microbe #2, the maximum specific oxygen-based growth rate for microbe #3, and the maximum specific nitrate-based growth rate for microbe #3 (1/T), respectively; $Y_o^{(1)}$, $Y_n^{(2)}$, $Y_o^{(3)}$, and $Y_n^{(3)}$ are the yield coefficient for microbe #1 utilizing oxygen, the yielding coefficient for microbe #2 utilizing nitrate, the yielding coefficient for

microbe #3 utilizing oxygen and nitrate, in mass of microbe per unit mass of substrate (M/M); $K_{so}^{(1)}$, $K_{so}^{(3)}$, $K_{sn}^{(2)}$, $K_{sn}^{(3)}$, $K_{po}^{(1)}$, $K_{po}^{(3)}$, $K_{pn}^{(2)}$, $K_{pn}^{(3)}$ are the retarded substrate saturation constants under aerobic conditions with respect to microbe #1, #3, the retarded substrate saturation constants under anaerobic conditions with respect to microbe #2, #3, the retarded nutrient saturation constants under aerobic conditions with respect to microbe #1, microbe #3, and the retarded nutrient saturation constants under anaerobic conditions with respect to microbe #2, microbe #3, respectively; $K_{o}^{(1)}$, $K_{o}^{(3)}$, $K_{n}^{(2)}$, $K_{n}^{(3)}$ are the retarded oxygen saturation constants under aerobic conditions with respect to microbe #1, microbe #3, and the retarded nitrate saturation constant under anaerobic conditions with respect to microbe #2 and microbe #3 (M/L^3), respectively. $\lambda_{o}^{(1)}$, $\lambda_{o}^{(3)}$, $\lambda_{n}^{(2)}$, and $\lambda_{n}^{(3)}$ are the microbial decay constants of aerobic respiration of microbe #1 and microbe #3, and the microbial decay constants of anaerobic respiration of microbe #2 and microbe #3 (1/T), respectively; $\gamma_{o}^{(1)}$, $\gamma_{o}^{(3)}$, $\gamma_{n}^{(2)}$, and $\gamma_{n}^{(3)}$ are the oxygen-use or nitrate-use for syntheses by microbe #1, microbe #2, or microbe #3, respectively; $\alpha_{o}^{(1)}$, $\alpha_{o}^{(3)}$, $\alpha_{n}^{(2)}$, and $\alpha_{n}^{(3)}$ are the oxygen-use or nitrate-use coefficient for energy by microbe #1, microbe #2, or microbe #3, respectively; $\Gamma_{o}^{(1)}$, $\Gamma_{o}^{(3)}$, $\Gamma_{n}^{(2)}$, and $\Gamma_{n}^{(3)}$ are the oxygen or nitrate saturation constants for decay with respect to microbe #1, microbe #2, or microbe #3 (M/L^3), respectively; and $\epsilon_{o}^{(1)}$, $\epsilon_{o}^{(3)}$, $\epsilon_{n}^{(2)}$, and $\epsilon_{n}^{(3)}$ are the nutrient-use coefficients for the production of microbe #1, microbe #2, or microbe #3 with respect to aerobic or anaerobic respiration, respectively.

Initial Conditions for Transport

$$C_s = C_{si}(x,z)$$

$$C_o = C_{oi}(x,z)$$

$$C_n = C_{ni}(x,z)$$

$$C_p = C_{pi}(x,z) \qquad in \quad R \qquad\qquad (2.17)$$

$$C_1 = C_{1i}(x,z)$$

$$C_2 = C_{2i}(x,z)$$

$$C_3 = C_{3i}(x,z)$$

Prescribed Concentration (Dirichlet) Boundary Conditions

$$C_s = C_{sd}(x_b,z_b,t)$$

$$C_o = C_{od}(x_b,z_b,t)$$

$$C_n = C_{nd}(x_b,z_b,t)$$

$$C_p = C_{pd}(x_b,z_b,t) \qquad on \quad B_d \qquad\qquad (2.18)$$

$$C_1 = C_{1d}(x_b,z_b,t)$$

$$C_2 = C_{2d}(x_b,z_b,t)$$

$$C_3 = C_{3d}(x_b,z_b,t)$$

<u>Variable Boundary Conditions</u>

$$\mathbf{n}\cdot(\mathbf{V}C_s - \theta\mathbf{D}\cdot\nabla C_s) = \mathbf{n}\cdot\mathbf{V}C_{sv}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_o - \theta\mathbf{D}\cdot\nabla C_o) = \mathbf{n}\cdot\mathbf{V}C_{ov}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_n - \theta\mathbf{D}\cdot\nabla C_n) = \mathbf{n}\cdot\mathbf{V}C_{nv}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_p - \theta\mathbf{D}\cdot\nabla C_p) = \mathbf{n}\cdot\mathbf{V}C_{pv}(x_b, z_b, t) \qquad \textit{if } \mathbf{n}\cdot\mathbf{V} \le 0 \qquad\qquad (2.19a)$$

$$\mathbf{n}\cdot(\mathbf{V}C_1 - \theta\mathbf{D}\cdot\nabla C_1) = \mathbf{n}\cdot\mathbf{V}C_{1v}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_2 - \theta\mathbf{D}\cdot\nabla C_2) = \mathbf{n}\cdot\mathbf{V}C_{2v}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_3 - \theta\mathbf{D}\cdot\nabla C_3) = \mathbf{n}\cdot\mathbf{V}C_{3v}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(-\theta\mathbf{D}\cdot\nabla C_s) = 0$$

$$\mathbf{n}\cdot(-\theta\mathbf{D}\cdot\nabla C_p) = 0$$

$$\mathbf{n}\cdot(-\theta\mathbf{D}\cdot\nabla C_n) = 0$$

$$\mathbf{n}\cdot(-\theta\mathbf{D}\cdot\nabla C_p) = 0 \qquad \textit{if } \mathbf{n}\cdot\mathbf{V} > 0 \qquad\qquad (2.19b)$$

$$\mathbf{n}\cdot(-\theta\mathbf{D}\cdot\nabla C_1) = 0$$

$$\mathbf{n}\cdot(-\theta\mathbf{D}\cdot\nabla C_2) = 0$$

$$\mathbf{n}\cdot(-\theta\mathbf{D}\cdot\nabla C_3) = 0$$

<u>Cauchy Boundary Conditions</u>

$$\mathbf{n}\cdot(\mathbf{V}C_s - \theta\mathbf{D}\cdot\nabla C_s) = q_{sc}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_o - \theta\mathbf{D}\cdot\nabla C_o) = q_{oc}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_n - \theta\mathbf{D}\cdot\nabla C_n) = q_{nc}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_p - \theta\mathbf{D}\cdot\nabla C_p) = q_{pc}(x_b, z_b, t) \qquad \text{on } B_c \qquad\qquad (2.20)$$

$$\mathbf{n}\cdot(\mathbf{V}C_1 - \theta\mathbf{D}\cdot\nabla C_1) = q_{1c}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_2 - \theta\mathbf{D}\cdot\nabla C_2) = q_{2c}(x_b, z_b, t)$$

$$\mathbf{n}\cdot(\mathbf{V}C_3 - \theta\mathbf{D}\cdot\nabla C_3) = q_{3c}(x_b, z_b, t)$$

$$\mathbf{n} \cdot (-\theta \mathbf{D} \cdot \nabla C_s) = q_{sn}(x_b, z_b, t)$$

$$\mathbf{n} \cdot (-\theta \mathbf{D} \cdot \nabla C_o) = q_{on}(x_b, z_b, t)$$

$$\mathbf{n} \cdot (-\theta \mathbf{D} \cdot \nabla C_n) = q_{nn}(x_b, z_b, t)$$

$$\mathbf{n} \cdot (-\theta \mathbf{D} \cdot \nabla C_p) = q_{pn}(x_b, z_b, t) \qquad \text{on} \quad B_n \qquad (2.21)$$

$$\mathbf{n} \cdot (-\theta \mathbf{D} \cdot \nabla C_1) = q_{1n}(x_b, z_b, t)$$

$$\mathbf{n} \cdot (-\theta \mathbf{D} \cdot \nabla C_2) = q_{2n}(x_b, z_b, t)$$

$$\mathbf{n} \cdot (-\theta \mathbf{D} \cdot \nabla C_3) = q_{3n}(x_b, z_b, t)$$

where C_{si}, C_{oi}, C_{ni}, C_{pi}, C_{1i}, C_{2i}, and C_{3i}, are the initial concentrations of substrate, oxygen, nitrogen, nutrient, microbe #1, #2, and #3; R is the region of interest; (x_b, z_b) is the spatial coordinate on the boundary; $\mathbf{n}$ is an outward unit vector normal to the boundary; C_{sd}, C_{od}, C_{nd}, C_{pd}, C_{1d}, C_{2d}, C_{3d}, and C_{sv}, C_{ov}, C_{nv}, C_{pv}, C_{1v}, C_{2v}, C_{3v}, are the prescribed concentrations of substrate, oxygen, nitrogen, nutrient, microbe #1, #2, and #3, on the Dirichlet boundary and the specified concentrations of water through the variable boundary, respectively; B_d, and B_v are the Dirichlet and variable boundaries respectively; q_{sc}, q_{oc}, q_{nc}, q_{pc}, q_{1c}, q_{2c}, q_{3c} and q_{sn}, q_{on}, q_{nn}, q_{pn}, q_{1n}, q_{2n}, q_{3n}, are the prescribed total flux and gradient flux of substrate, oxygen, nitrogen, nutrient, microbe #1, #2, and #3 through the Cauchy and Neumann boundaries B_c and B_n, respectively.

2.2 Numerical Approximation

Flow Equation

The pressure head can be approximated with Eq. (2.22) by the finite element method.

$$h \approx \sum_{j=1}^{N} h_j(t) N_j(x, z) \qquad (2.22)$$

where N is the total number of nodes in the region and N_j and h_j are the basis function and the amplitude of

h respectively, at nodal point j. Substituting Eq. (2.22) into Eq. (2.1) and choosing Galerkin finite element

method, the governing flow equation can be approximated to the following.

$$\left[\int_R N_i \frac{\rho}{\rho_w} F N_j dR\right]\frac{dh_j}{dt} + \left[\int_R (\nabla N_i)\cdot\mathbf{K}\cdot(\nabla N_j)dR\right]h_j =$$

$$\int_R N_i \frac{\rho^*}{\rho_w}(\text{or} - \frac{\rho}{\rho_w})qdR - \int_R (\nabla N_i)\cdot\mathbf{K}\cdot(\frac{\rho}{\rho_w}\nabla z)dR + \int_B \mathbf{n}\cdot\mathbf{K}\cdot(\nabla h + \frac{\rho}{\rho_w}\nabla z)N_i dB, \qquad (2.23)$$

$$\text{where } i=1,2,...,N. \qquad F = \frac{d\theta}{dt}$$

Equation (2.23) written in matrix form is:

$$[M]\{\frac{dh}{dt}\} + [S]\{h\} = \{G\} + \{Q\} + \{B\} \qquad (2.24)$$

where {dh/dt} and {h} are the column vectors containing the values of dh/dt and h respectively at all nodes;

[M] is the mass matrix resulting from the storage term; [S] is stiffness matrix resulting from the action of

conductivity; {G}, {Q} and {B} are the load vectors from the gravity force, internal source/sink, and boundary

conditions, respectively. The matrices, [M] and [S] are given by

$$M_{ij} = \sum_{e\in M_e} \int_{R_e} N_\alpha^e \frac{\rho}{\rho_w} F N_\beta^e dR \qquad (2.25)$$

$$S_{ij} = \sum_{e\in M_e} \int_{R_e} \nabla\left(N_\alpha^e\right)\mathbf{K}\cdot\nabla\left(N_\beta^e\right)dR \qquad (2.26)$$

where

R_e = the region of element e,

M_e = the set of elements that have a local side α-β coinciding with the global side i-j,

N_α^e = the α-th local basis function of element e.

Similarly the load vectors {G}, {Q} and {B} are given by

$$G_i = -\sum_{e \in M_e} \int_{R_e} (\nabla N_\alpha^e) \cdot \mathbf{K} \cdot \frac{\rho}{\rho_w} \nabla z dR \tag{2.27}$$

$$B_i = -\sum_{e \in N_{se}} \int_{B_e} N_\alpha^e \cdot [-\mathbf{K} \cdot (\nabla h + \frac{\rho}{\rho_w} \nabla z)] dB \tag{2.28}$$

$$Q_i = \sum_{e \in M_e} \int_{R_e} N_\alpha^e \frac{\rho^*}{\rho_w} (or - \frac{\rho}{\rho_w}) q dR \tag{2.29}$$

where

B_e = the length of boundary segment e,

N_{se} = the set of boundary segments that have a local node α coinciding with the global node i.

The following steps illustrate the incorporation of boundary conditions into matrix equation by finite element

method.

For the Cauchy boundary condition given by Eq. (2.7), we simply substitute Eq. (2.7) into Eq. (2.28)

to yield a boundary-element column vector $\{B_c^e\}$ for a Cauchy segment:

$$\{B_c^e\} = \{q_c^e\} \tag{2.30}$$

where $\{q_c^e\}$ is the Cauchy boundary flux vector given by

$$q_{c\alpha}^e = -\int_{B_e} N_\alpha^e \frac{\rho}{\rho_w} q_c dB , \qquad \alpha = 1 \ or \ 2 \tag{2.31}$$

The Cauchy boundary flux vector represents the normal fluxes through the two nodal points of the segment

B_e on B_c.

For the Neumann boundary condition given by Eq. (2.6), we substitute Eq. (2.6) into Eq. (2.28) to yield a boundary-element column vector $\{B_n^e\}$ for a Neumann segment:

$$\{B_n^e\} = \{q_n^e\} \tag{2.32}$$

where $\{q_{n\alpha}^e\}$ is the Neumann boundary flux vector given by:

$$q_{n\alpha}^e = \int_{B_e}\left(N_\alpha^e \mathbf{n}\cdot\mathbf{K}\cdot\frac{\rho}{\rho_w}\nabla z - N_\alpha^e q_n \right) dB \ ; \quad \alpha = 1 \text{ or } 2 \tag{2.33}$$

which is independent of the pressure head.

The implementation of the variable-type boundary condition is more involved. During the iteration of boundary conditions on the variable boundary, one of Eqs.(2.8a) through (2.8e) is used at a node. If either Eq.(2.8b) or (2.8e) is used, we substitute it into Eq. (2.28) to yield a boundary element column vector $\{B_v^e\}$ for a variable boundary segment:

$$\{B_v^e\} = \{q_v^e\} \tag{2.34}$$

where $\{q_v^e\}$ is the variable boundary flux given by:

$$q_{v\alpha}^e = -\int_{B_e} N_\alpha^e \frac{\rho}{\rho_w} q_p dB, \quad \text{or } q_{v\alpha} = -\int_{B_e} N_\alpha^e \frac{\rho}{\rho_w} q_e dB \ ; \quad \alpha = 1 \text{ or } 2 \tag{2.35}$$

Assembling over all Neumann, Cauchy, and variable boundary segments, we obtain the global boundary column vector $\{B\}$ as:

$$\{B\} = \{q\} \tag{2.36}$$

in which

$$\{q\} = \sum_{e\in N_{ne}} \{q_n^e\} + \sum_{e\in N_{ce}} \{q_c^e\} + \sum_{e\in N_{ve}} \{q_v^e\} \tag{2.37}$$

16

where N_{ne}, N_{ce}, and N_{ve} are the number of Neumann boundary segments, Cauchy boundary segments, and variable boundary segments with flux conditions imposed on them, respectively. The boundary flux {B} given by Eqs. (2.36) and (2.37) should be added to the right hand side of Eq. (2.24).

At nodes where Dirichlet boundary conditions are applied, an identity equation is generated for each node and included in the matrices of Eq. (2.24). The Dirichlet nodes include the nodes on the Dirichlet boundary and the nodes on the variable boundary to which either Eq.(2.8a), (2.8c), or (2.8d) is applied.

Transport Equation

To simplify the notation, the subscript s, o, n, p, 1, 2, or 3 in Eq. (2.9) to (2.15) will be dropped for the development of numerical algorithm in this section. Since the hybrid Lagrangian-Eulerian approach is used to simulate Eq. (2.9) to (2.15), it is written in the Lagrangian-Eulerian form as

$$(\theta + \rho_b K_d)\frac{DC}{Dt} = \nabla \cdot (\theta \mathbf{D} \cdot \nabla C) - \Lambda(\theta C + \rho_b K_d) + QC_{in} - \frac{\rho^*}{\rho}QC + \frac{\rho_w}{\rho}\mathbf{V} \cdot \nabla(\frac{\rho}{\rho_w})C$$
$$- f(C_1, C_2, C_3, C_s, C_o, C_n, C_p)C + g(C_1, C_2, C_3, C_s, C_o, C_n, C_p)C \tag{2.38}$$

$$\mathbf{V}^* = \frac{\mathbf{V}}{\theta + \rho_b K_d} \tag{2.39}$$

where $f(C_1, C_2, C_3, C_s, C_o, C_n, C_p)$ is a microbial-chemical interaction function and $g(C_1, C_2, C_3, C_s, C_o, C_n, C_p)$ is a microbial growth function . Applying the Galerkin finite element method to Eq. (2.9) through Eq. (2.15), we obtain

$$[M]\{\frac{DC}{Dt}\} + ([A] + [D] + [K] + [B])\{C\} = \{Q\} + \{B\} \tag{2.40}$$

where {C} is a vector whose components are the concentrations at all nodes, {DC/Dt} is the time derivative of {C} with respect to time, [M] is the mass matrix associated with the time derivative term, [A] is the stiffness

17

matrix associated with the velocity term which is only computed when steady-state is considered, [D] is the stiffness matrix associated with the dispersion term, [K] is the stiffness matrix associated with the decay term and microbial-chemical interaction, [B] is the stiffness matrix resulting from boundary conditions, {Q} is the load vector associated with all source/sink terms, and {B} is the load vector associated with boundary conditions. These matrices and vectors are given as follows.

$$M_{ij} = \sum_{e \in M_e} \int_{R_e} N_\alpha^e (\theta + \rho_b K_d) N_\beta^e dR \tag{2.41a}$$

$$A_{ij} = \sum_{e \in M_e} \int_{R_e} W_\alpha^e \mathbf{V} \cdot \nabla N_\beta^e dR \tag{2.41b}$$

$$D_{ij} = \sum_{e \in M_e} \int_{R_e} \nabla N_\alpha^e \cdot \mathbf{D} \cdot \nabla N_\beta^e dR \tag{2.41c}$$

$$K_{ij} = \sum_{e \in M_e} \int_{R_e} N_\alpha^e \left[\Lambda(\theta + \rho_b K_d) + f(C_1, C_2, C_3, C_s, C_o, C_n, C_p) + \frac{\rho^*}{\rho} Q_{in} - \frac{\rho_w}{\rho} \mathbf{V} \cdot \nabla \frac{\rho}{\rho_w} \right] N_\beta^e dR \tag{2.41d}$$

$$B_{ij} = - \left[\sum_{B_e \in B_v^+} \int_{B_e} N_\alpha^e (\mathbf{n} \cdot \mathbf{V}) N_\beta^e dB + \sum_{B_e \in B_c} \int_{B_e} N_\alpha^e (\mathbf{n} \cdot \mathbf{V}) N_\beta^e dB \right] \tag{2.41e}$$

$$Q_i = \sum_{e \in M_e} \int_{R_e} N_\alpha^e Q_{in} C_{in} dR + \sum_{e \in M_e} \int_{R_e} N_\alpha^e g(C_1, C_2, C_3, C_s, C_o, C_n, C_p) dR \tag{2.41f}$$

$$B_i = - \left[\sum_{B_e \in B_v^+} \int_{B_e} N_\alpha^e (\mathbf{n} \cdot \mathbf{V}) C_v dB + \sum_{B_e \in B_c} \int_{B_e} N_\alpha^e q_c dB + \sum_{B_e \in B_n} \int_{B_e} N_\alpha^e q_n dB \right] \tag{2.41g}$$

where B_v^+ is that part of variable boundary for which the flow is directed into the region, C_v is the concentration of the incoming fluid through the variable boundary segment B_v^+, and B_c, B_n are the Cauchy and Neumann boundary segments.

18

The numerical algorithm to solving this partial differential equation is a modified Lagrangian-Eulerian method with adapted zooming and peak capturing (LEZOOMPC). Basically, LEZOOMPC is a modified method of the zoomable hidden fine-mesh approach (LEZOOM) (Yeh, 1990) and exact peak capturing and oscillation-free scheme (EPCOF) (Yeh et al., 1992) to solve advection-dispersion transport equations. The detail algorithm of the LEZOOMPC can be found elsewhere (Cheng, et. al., 1995). Two of the most important terminologies used in the LEZOOMPC algorithm are the rough element and smooth element. Two terms need to be defined, which shall used throughout this document. A smooth element is defined as an element within which any physical quantity at all points can be interpolated with its node values to within error tolerance. A rough element is defined as an element within which there exists at least one point for which the physical quantity cannot be interpolated with its node values to within error tolerance.

Figure 2.1 is the basic concept structure of solving transport of 2DFATMIC. It contains two main steps, Lagrangian and Eulerian steps. First, the concentrations at last time step t^n, C^n's, are known quantities at the beginning of this time step. Second (GNTRAK module), we compute the Lagrangian concentration C_i^*'s at global nodes using the backward node tracking as

$$C_i^* = \sum_{j=1}^{N+N_n} C_j^n N_j(x_i^*, z_i^*), \quad i=1,2,..,N \tag{2.42}$$

in which

$$x_i^* = x_i - \int_{t_n}^{t_{n+1}} V_x dt$$

$$z_i^* = z_i - \int_{t_n}^{t_{n+1}} V_z dt \quad i=1,2,..,N \tag{2.43}$$

where $N_j(x_i^*)$ is the base function associated with node x_j evaluated at x_i^*; V_x and V_z are the velocities along x- and y- direction, respectively.

19

Third (HPTRAK modules), all the activated fine grid nodes and the global nodes are forwardly tracked to obtain the Lagrangian concentration C_j^f by the following equations:

$$C_j^f = C\left(x_j^f, z_j^f, t_{n+1}\right) = C\left(x_j, z_j, t_n\right) = C_j^n \qquad j=1,2,\ldots,N+N_n \qquad (2.44)$$

in which

$$x_j^f = x_j + \int_{t_n}^{t_{n+1}} V_x dt \quad , \quad z_j^f = z_j + \int_{t_n}^{t_{n+1}} V_z dt \qquad (2.45)$$

It should be noted that C_j^f's are exact if C_j^n's are exact and Eq.(2.45) is integrated exactly.
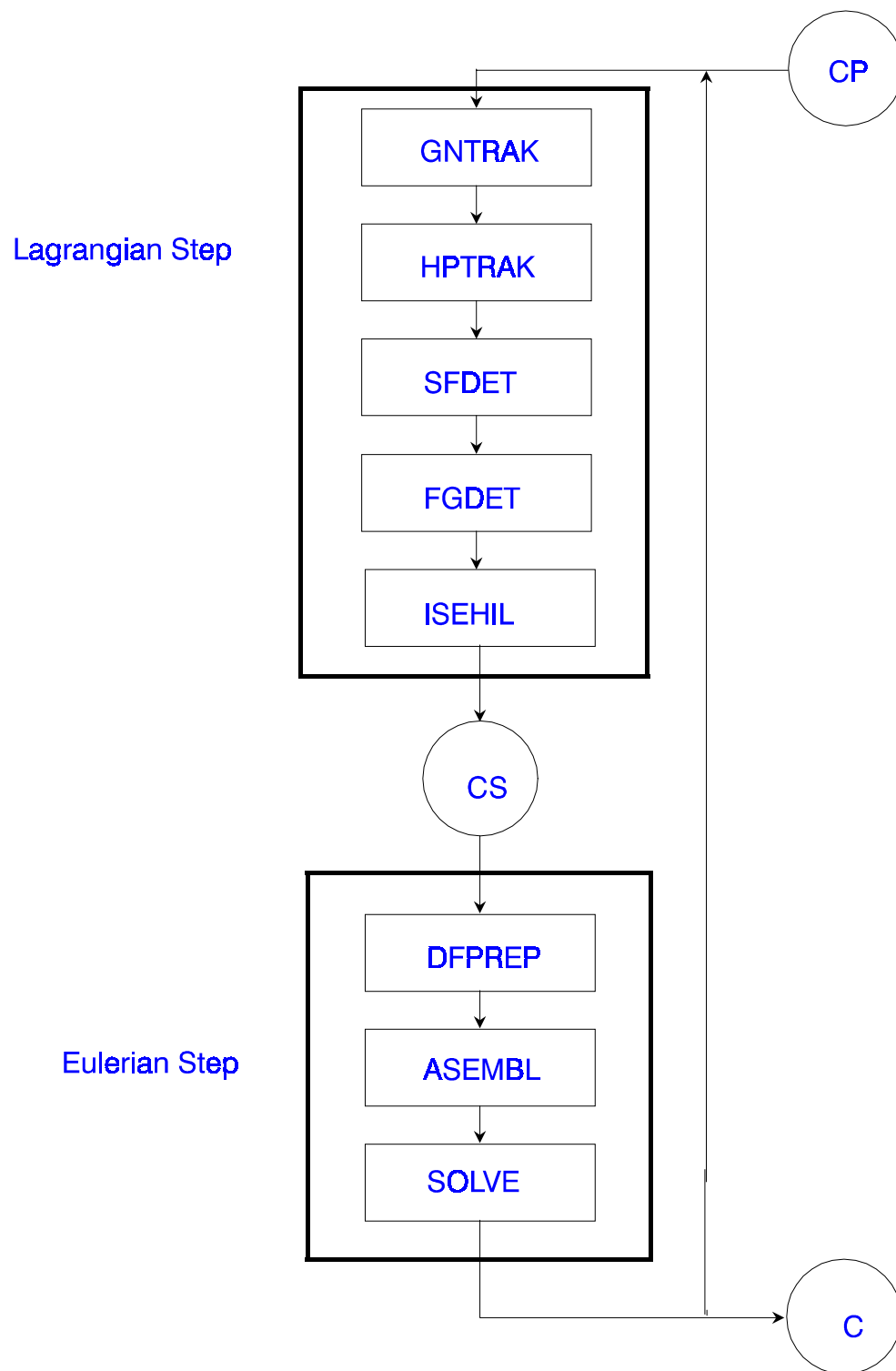
Figure 2.1 The basic structure for coding transport part of 2DFATMIC

Fourth (SFDET and FGDET modules), we determine if an element is a rough element in SFDET module (Yeh, 1990) based on the prescribed error tolerance. The criterion is shown as follows:

$$\left| \left(C_j^a - C_j^f \right) / C_j^f \right| \leq \epsilon_1$$
$$\left| \left(C_j^a - C_j^f \right) / C_M^f \right| \leq \epsilon_2$$

(2.46)

where C_j^a is the approximate value determined from finite element interpolation, ϵ_1, ϵ_2 are two error tolerances and C_M^f is the maximum concentration of C_j^f's. FGDET module generates the regular fine-mesh points in every rough element resulted from SFDET module.

Fifth (ISEHIL module), if the element is a smooth element, all forward-tracked nodes for dispersion computation at the present time t_{n+1} and advection computation at next time t_{n+2} are removed. Otherwise, a number of subelements, which is determined by the users, are used to refine the rough element. The indices of subelements are stored in the ISE array. In addition to regular refinement, this module also captures the highest and lowest concentrations within each subelement. This demonstrates the idea called adapted zooming and peak capturing. The above five steps form the Lagrangian computation of advective transport.

The next three steps (DFPREP, ASEMBL, and SOLVE modules) are Eulerian steps to solve the dispersion matrix equation. The module DFPREP prepares all the needed information for assembling the subelemental matrices, which are zoomed in the Eulerian step. The inclusion of this module gets rid of the inaccuracy due to the dispersion calculation. In this module, the number of subelements generated in each global element is determined by the user and may be different from that in the module FGDET. If the element is a smooth element, no fine grids are imbedded. If the element is rough, it is zoomed and refined with the Delauney triangulation. At the end of this module, the nodal connection information of each point has to be prepared to compose the matrix and solve the matrix equation.

The module ASEMBL is designed to yield the following element matrix equation

$$\{A^e\}\{C^e\} = \{R^e\}$$

(2.47)

22

which is based on Eqs. (2.9) to (2.15), Where $[A^e]$ is the element coefficient matrix, $\{C^e\}$ is the unknown vector of the concentration, and $\{R^e\}$ is the element load vector. Element e can be a global element or a subelement generated in a rough region by DFPREP. Then, this module assembles all the element matrix equations to a global matrix equation, which will be solved by a diffusion solver.

The module SOLVE solves the assembled global matrix equation by a direct solver or pointwise iteration method. If the diffusion zoomed approach is activated in the Eulerian step, the latter is selected forcefully.

At the very end of this time step, i.e. at t_{n+1}, the concentrations at all fine nodal points (including regularly refined nodes and peak/valley nodes) generated in Lagrangian step, are obtained by finite element interpolations as follows:

$$C_k^{n+1} = C_k^f + \sum_{i=1}^{N} \left(C_i^{n+1} - C_i^*\right) N_i\left(x_k^f\right), \quad k \in \text{All Fine Nodal Points} \tag{2.48}$$

2.3 Description of 2DFATMIC Subroutines

2DFATMIC consists of a MAIN program and 77 subroutines. The MAIN is utilized to specify the sizes of all arrays. The control and coordinate activity are performed by the subroutine HTMICH. Figure 2.2 shows the structure of the program. The functions of these subroutines are described below.

Subroutine HTMICH

The subroutine HTMICH controls the entire sequence of operations, a function generally performed by the MAIN program. It is, however, preferable to keep a short MAIN and several subroutines with variable short allocation. This makes it possible to place most of the FORTRAN deck on a permanent file and to deal with a specific problem without making changes in array dimensions throughout all subroutines.

The flow of data input for the model is also anchored by the HTMICH. The subroutine RDATIO is called to read the geometric and material data. HTMICH then calls subroutine IBE2D to generate IBE array

23

which stores the index of boundary elements. The source/sink data and boundary conditions for flow and transport are read by subroutines FDATIO and TDATIO, respectively. The microbial-chemical reaction parameters are also read in subroutine TDATIO. Figure 2.2 shows the flow chart of this subroutine.

Depending on the combinations of the parameters KSSh, KSSt, NTI, and IMOD, the subroutine HTMICH will perform either the steady-state flow and/or transport computations only, or the transient state flow and/or transport computations using the flow and/or transport steady-state solution as the initial conditions, or the transient flow and/or transport computation using user-supplied initial conditions.

HTMICH calls the subroutines ESSFCT, WSSFCT, CBVFCT, NBVFCT, VBVFCT, and DBVFCT to obtain sources/sinks and boundary values; subroutine SPROP to obtain the relative hydraulic conductivity, water capacity, and moisture content from the pressure head; subroutine VELT to compute Darcy's velocity; subroutine FSFLOW to calculate flux through all types of boundaries and water accumulated in the media; subroutine FPRINT to print out the results; and subroutine FSTORE to store the flow variables for plotting; subroutine HMCHYD to perform the flow computations; subroutine FLUX to compute material flux; subroutine AFABTA to obtain upstream weighting factors based on velocity and dispersivity; subroutine TSFLOW to calculate material fluxes through all types of boundaries and the amount of water accumulated in the media; subroutine TPRINT to print out the transport computation results; subroutine TSTORE to store the transport computation results for plotting; subroutine THNODE to compute the value of the moisture content plus bulk density times distribution coefficient; subroutine DISPC to compute the dispersion coefficients; subroutine HMCTRN to perform the transport computations.
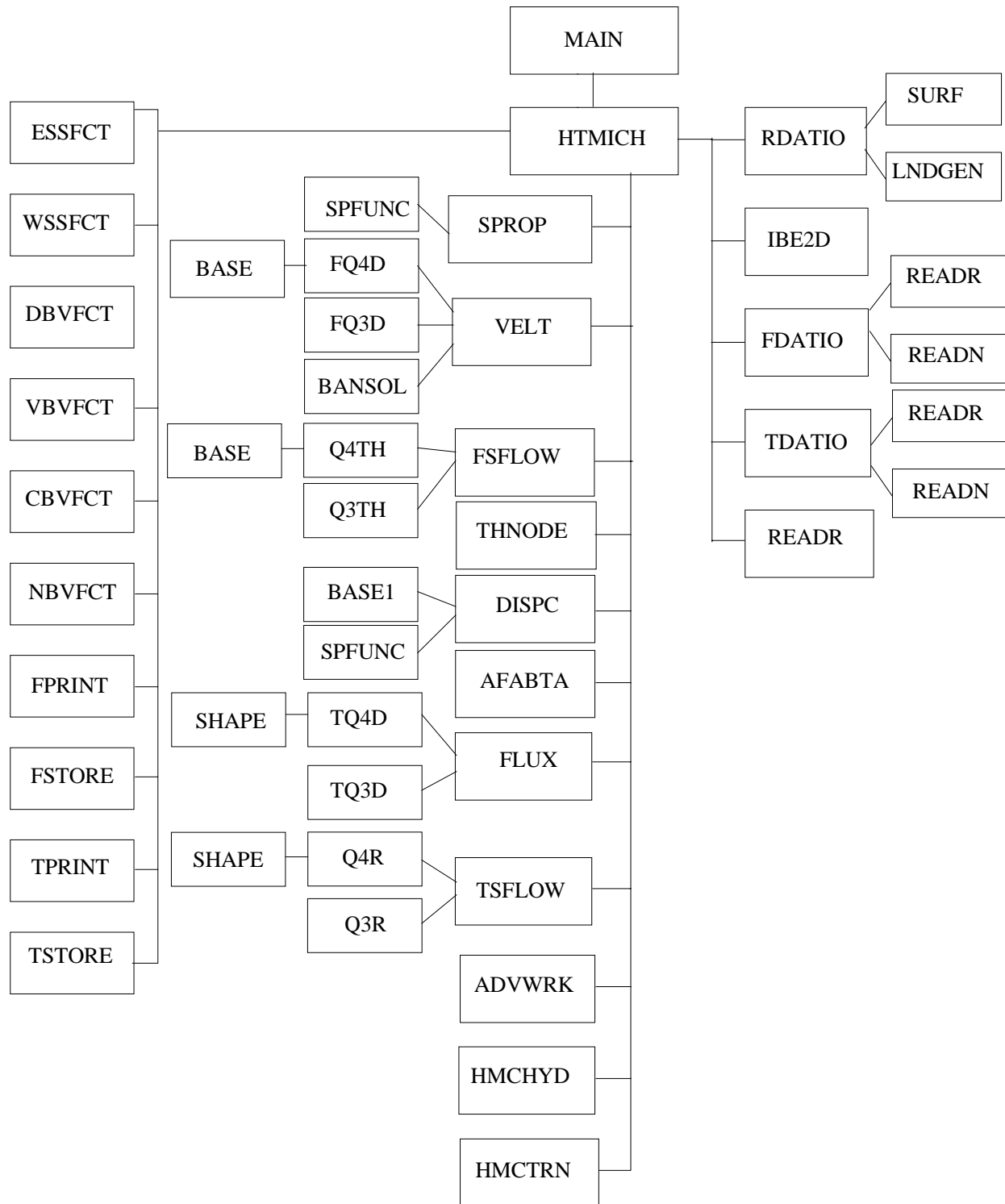
Figure 2.2 Program structure of 2DFATMIC (MAIN)

25

Subroutine RDATIO

  Subroutine RDATIO reads data set 2 to data set 6 described by Appendix A. It also prints all the input information, and calls subroutine READR and READN, respectively, to read and/or automatically generate real and integer numbers; subroutine LNDGEN to generate two pointer arrays (one is the node stencil and the other is the connecting elements to any node); subroutine SURF to identify the boundary segments and boundary nodes.

Subroutine READR

This subroutine is called by the subroutines RDATIO, FDATIO, TDATIO, and HTMICH to read some real numbers and automatically generate other real numbers if required. Automatic generation of regular patterned data is built into the subroutine (see Appendix A).

Subroutine READN

This subroutine is also called by the subroutines RDATIO, FDATIO, and TDATIO to generate read some integers and automatically generate other integers if required (see Appendix A).

Subroutine LNDGEN

This subroutine is called by the controlling subroutine HMCTRN to preprocess the two pointer arrays. One is the global node connectively (stencil) LRN(J,N), where LRN(J,N) is the global node number of J-th node connected to the global node N. This pointer array serves two purposes: (1) to assemble the global matrix in compressed form when the pointwise iteration method is used, (2) to speed up locating fictitious particles in the Lagrangian step. The other pointer array is the global elements connected to nodes LRL(K,N), where LRL(K,N) is the global element number of the K-th elements connected to the global node N. These two pointer arrays are generated based on the element connectivity IE(M,J). Here, IE(M,J) is the global node number of J-th node of element M. These arrays are used in particle tracking, i.e., subroutine GNTRAK and HPTRAK.

Subroutine SURF

Subroutine SURF identifies the boundary sides, sequences the boundary nodes, and computes the directional cosine of the surface sides. The mapping from boundary nodes to global nodes are stored in NPBB(I) (where NPBB(I) is the global node number of i-th boundary node). The element number associated with the boundary sides are stored in ISB array. The length and directional cosines for each side are stored in DLCSB array, respectively. The local and global nodal number of two nodes of each side are also stored in ISB. The information contained in NPBB, ISB, and DLCSB along with the number of boundary nodes and the number of boundary sides are returned to subroutine RDATIO for other users.

Subroutine IBE2D

The subroutine IBE2D is used to generate the index of boundary element stored in IBE array. If IBE(M) = 0, it means no boundary element side in the M-th element. If IBE(M)=12, the element side connected by local node number 1 and 2 is the boundary element side of the M-th element.

Subroutine FDATIO

The subroutine FDATIO is called by the program HTMICH and to control the input of initial condition, sources/sinks, and boundary conditions, in time and space, assigned to each specified node/element for flow simulations. Users need to give the initial values, source/sink profiles, and boundary profiles, to specify the global node/element numbers of the boundary, and to assign boundary profile type to each node/element for flow simulations. The source/sink type for each node/element is also assigned in this subroutine according to the data given by the users.

Subroutine TDATIO

The subroutine TDATIO is called by the program HTMICH and to control the input of initial condition, sources/sinks, and boundary conditions, in time and space, assigned to each specified node/element for transport simulations. Users need to give the initial values, source/sink profiles, and boundary profiles, to specify the global node/element numbers of the boundary, and to assign boundary profile type to each

27

node/element for transport simulations. The source/sink type for each node/element is also assigned in this subroutine according to the data given by the users.

Subroutine ESSFCT

This subroutine is called by the subroutine HTMICH to compute the elemental source strength. It uses linear interpolation of the tabular data or it computes the value with an analytical function. If the latter option is used, the user must supply the function to this subroutine.

Subroutine WSSFCT

This subroutine is called by the subroutine HTMICH to compute the well source strength. It uses the linear interpolation of the tabular data or it computes the value with analytical function. If the latter option is used, the user must supply the function into this subroutine.

Subroutine VBVFCT

This subroutine is called by the subroutine HTMICH to compute the variable boundary values. It uses linear interpolation of the tabular data or it computes the value with an analytical function. If the latter option is used, the user must supply the function to this subroutine.

Subroutine DBVFCT

This subroutine is called by the subroutine HTMICH to compute the Dirichlet boundary values. It uses linear interpolation of the tabular data or it computes the value with an analytical function. If the latter option is used, the user must supply the function to this subroutine.

Subroutine CBVFCT

This subroutine is called by the subroutine HTMICH to compute the Cauchy fluxes. It uses linear interpolation of the tabular data or it computes the value with an analytical function. If the latter option is used, the user must supply the function to this subroutine.

Subroutine NBVFCT

This subroutine is called by the subroutine HTMICH to compute the Neumann fluxes. It uses linear

interpolation of the tabular data or it computes the value with an analytical function. If the latter option is used, the user must supply the function to this subroutine.

Subroutine FSFLOW

This subroutine is used to compute the fluxes through various types of boundaries and the increasing rate of water content in the region of interest. The function of FRATE(7) is to store the flux through the whole boundary enclosing the region of interest. It is given by

$$FRATE(7) = \int_B (V_x n_x + V_z n_z) dB , \qquad (2.49)$$

where B is the global boundary of the region of interest; V_x and V_z are Darcy's velocity components; and n_x and n_z are the directional cosines of the outward unit vector normal to the boundary B. FRATE(1) through FRATE(5) store the flux through Dirichlet boundary B_D, Cauchy boundary B_C, Neumann boundary B_N, the seepage/evapotranspiration boundary B_s, and infiltration boundary B_r, respectively, and are given by

$$FRATE(1) = \int_{B_d} (V_x n_x + V_z n_z) dB , \qquad (2.50a)$$

$$FRATE(2) = \int_{B_c} (V_x n_x + V_z n_z) dB , \qquad (2.50b)$$

$$FRATE(3) = \int_{B_n} (V_x n_x + V_z n_z) dB , \qquad (2.50c)$$

$$FRATE(4) = \int_{B_s} (V_x n_x + V_z n_z) dB , \qquad (2.50d)$$

$$FRATE(5) = \int_{B_r} (V_x n_x + V_z n_z) dB , \qquad (2.50e)$$

29

FRATE(6), which is related to the numerical loss, is given by

$$\text{FRATE}(6) = \text{FRATE}(7) - \sum_{I=1}^{5} \text{FRATE}(I) \qquad (2.51)$$

FRATE(8) and FRATE(9) are used to store the source/sink and increased rate of water within the media, respectively:

$$\text{FRATE}(8) = -\int_{R} \frac{\rho^{*}}{\rho_{w}} q dR \ , \qquad (2.52)$$

and

$$\text{FRATE}(9) = \int_{R} \frac{\rho}{\rho_{w}} \frac{d\theta}{dh} \frac{\partial h}{\partial t} dR \ , \qquad (2.53)$$

where R is the region of interest. If there is no numerical error in the computation, the following equation should be satisfied:

$$\text{FRATE}(9) = -[\text{FRATE}(7) + \text{FRATE}(8)] \qquad (2.54)$$

and FRATE(6) should be equal to zero. Equation (2.54) simply states that the negative rate of water going out from the region through the entire boundary and due to a source/sink is equal to the rate of water accumulated in the region.

Subroutine Q4TH and Q3TH

These subroutines are used to compute the contribution of the increasing rate of the water content from an element e

$$\text{QTHP} = \int_{R_{e}} \frac{\rho}{\rho_{w}} \frac{d\theta}{dh} \frac{\partial h}{\partial t} dR \ , \qquad (2.55)$$

The computation of the above integration is straightforward. Q4TH is used to evaluate the integral for quadrilateral elements and Q3TH for triangular elements.

Subroutine FPRINT

This subroutine is used to line-print the flow variables. These include the fluxes through variable boundary surfaces, the pressure head, total head, moisture content, and Darcy's velocity components.

Subroutine FSTORE

This subroutine is used to store the flow variables on Logical Unit 11. It is intended to be used for plotting. The information stored includes region geometry, subregion data, and hydrological variables such as pressure head, total head, moisture content, and Darcy's velocity components.

Subroutine TPRINT

This subroutine is used to line-print the simulation results of the contaminant transport. These include the material flux components and the concentration at each node.

Subroutine TSTORE

This subroutine is used to store the simulation results on Logical Unit 12. It is intended for plotting purposes. The information stored includes region geometry, concentrations, and material flux components at all nodes for any desired time step.

Subroutine ADVWRK

This subroutine is called by HTMICH to prepare all the working arrays to be further used in ELETRK. This subroutine is called only when the transient simulation is required.

Subroutine HMCHYD

HMCHYD calls subroutine SPROP to obtain the relative hydraulic conductivity, water capacity, and moisture content from the pressure head; subroutine VELT to compute Darcy's velocity; subroutine BCPREP to determine if a change of boundary conditions is required; subroutine FASEMB to assemble the element

31

matrices over all elements; subroutine FBC to implement the boundary conditions; subroutine BANSOL or

Subroutine PISS to solve the linear matrix equations. Figure 2.3 shows the flow chart of this subroutine.
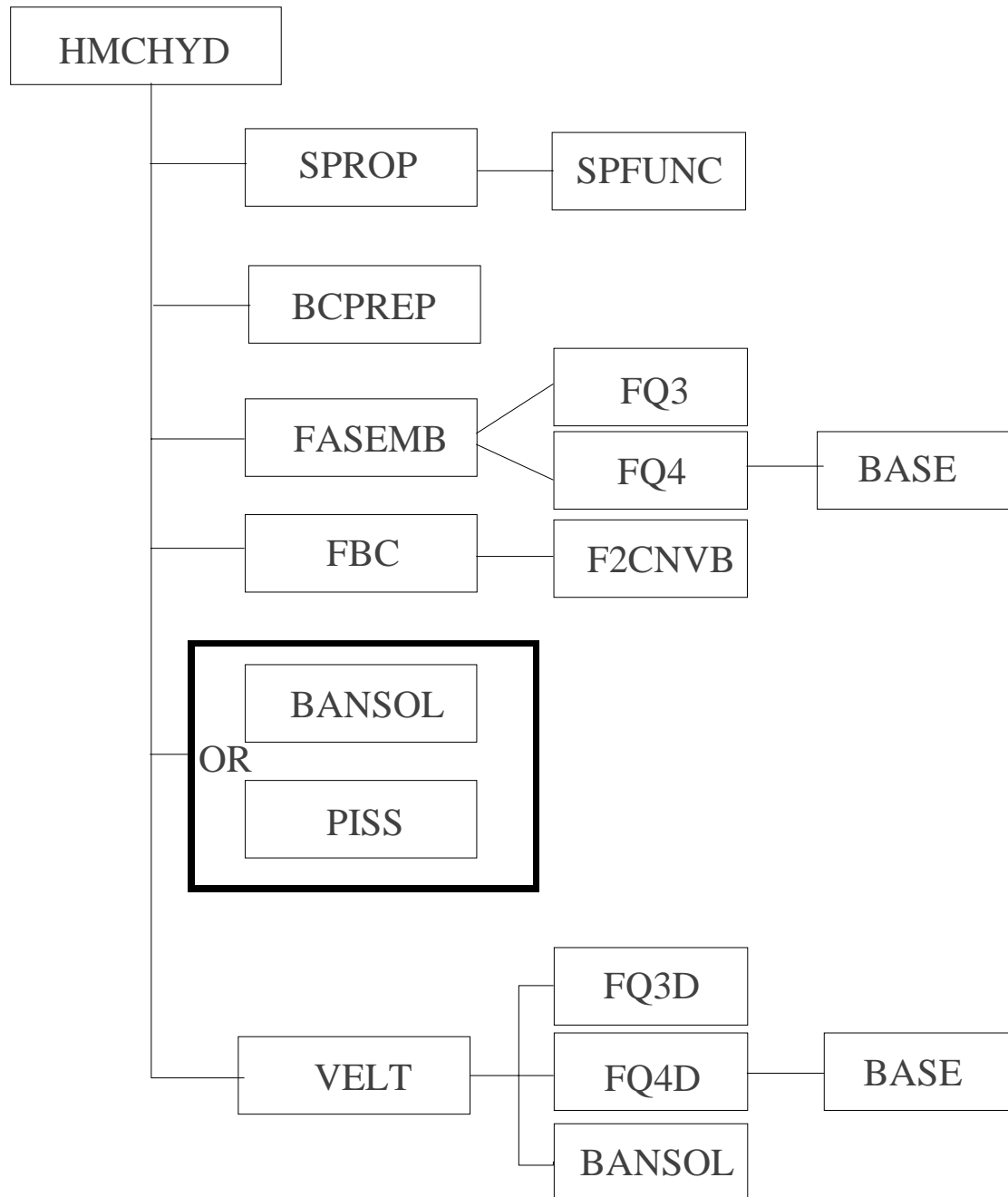
Figure 2.3 Program Structure of 2DFATMIC (Flow Part)

Subroutine SPROP

This subroutine calculates the values of moisture content, relative hydraulic conductivity, and the water capacity. This subroutine calls subroutine SPFUNC to calculate the soil property function by either tabular input or analytical functions.

Subroutine SPFUNC

This subroutine calculates the soil property function by either tabular input or analytical functions. When analytical functions are used, the users must supply the functional form and modify this subroutine.

Subroutine BCPREP

This subroutine is called by HMCHYD to prepare the infiltration-seepage boundary conditions during a rainfall period or the seepage-evapotranspiration boundary conditions during non-rainfall periods. It decides the number of nodal points on the variable boundary to be considered as Dirichlet or Cauchy points. It computes the number of points that change boundary conditions from ponding depth (Dirichlet types) to infiltration (Cauchy types), or from infiltration to ponding depth, or from minimum pressure (Dirichlet types) to infiltration during rainfall periods. It also computes the number of points that change boundary conditions from potential evapotranspiration (Cauchy types) to minimum pressure, or from ponding depth to potential evapotranspiration, or from minimum pressure to potential evapotranspiration during non-rainfall periods. Upon completion, this subroutine returns the Darcy flux, infiltration/potential evapotranspiration rate, the ponding depth nodal index, the flux-type nodal index, the minimum pressure nodal index, which are stored in RSVAB array, and the number of nodal points (NCHG) that have changed boundary conditions.

Subroutine FASEMB

This subroutine calls FQ4 or FQ3 to evaluate the element matrices. It then sums over all element matrices to form a global matrix equation governing the referenced pressure head at all nodes.

Subroutine FQ4 and FQ3

These subroutines are called by the subroutine FASEMB to compute the element matrix given by

$$QA(I,J) = \int\limits_{R_e} N_i^e \frac{\rho}{\rho_w} F N_j^e dR \; , \tag{2.56a}$$

$$QB(I,J) = \int\limits_{R_e} (\nabla N_i^e) \cdot \mathbf{K_s} K_r \cdot (\nabla N_j^e) dR \; , \tag{2.56b}$$

where F is the soil property function. Subroutine FQ4 and FQ3 also calculate the element load vector given by

$$RQ(I) = \int\limits_{R_e} [(\nabla N_i^e) \cdot \mathbf{K_s} K_r \cdot (\frac{\rho}{\rho_w} \nabla z) - N_i^e \frac{\rho^*}{\rho_w} (\text{or} - \frac{\rho}{\rho_w}) q] dR \; , \tag{2.56c}$$

where q is the source/sink. Subroutine FQ4 is to compute Eq.(2.56a) to Eq.(2.56c) for quadrilateral elements, while FQ3 is to compute those for triangular elements.

Subroutine BASE

This subroutine is called by subroutines FQ4D, Q4TH and FQ4 to evaluate the value of the base function at a Gaussian point. The computation is straightforward.

Subroutine FBC

This subroutine incorporates Dirichlet, Cauchy, Neumann, and variable boundary conditions. For a Dirichlet boundary condition, an identity algebraic equation is generated for each Dirichlet nodal point. Any other equation having this nodal variable is modified accordingly to simplify the computation. For a Cauchy boundary, the integration of Eq. (2.31) is obtained in subroutine FBC, and the result is added to the load vector. For a Neumann boundary, the integrations of both the Neumann and gravity fluxes (Eq.(2.33)) are added to the load vector. The subroutine FBC also implements the variable boundary conditions. First, it checks all infiltration-evapotranspiration-seepage points, identifying any of them that are Dirichlet points. If there are Dirichlet points, the method of incorporating Dirichlet boundary conditions mentioned above is used. If a given point is not the Dirichlet point, the point is bypassed. Second, it checks all rainfall-evaporation-

seepage points again to see if any of them is a Cauchy point. If it is a Cauchy point, then the computed flux by infiltration or potential evapotranspiration (Eq.(2.35)) is added to the load vector. If a given point is not a Cauchy point, it is bypassed. Because the infiltration-evaporation-seepage points are either Dirichlet or Cauchy points, all points are taken care of in this manner.

Subroutine F2CNVB

This subroutine is called by the subroutines FBC to compute the surface node flux of the type

$$RQ(I) = -\int_{B_{ve}} N_i \frac{\rho}{\rho_w} q_v dB \qquad (2.57a)$$

$$RQ(I) = -\int_{B_{ce}} N_i \frac{\rho}{\rho_w} q_c dB \qquad (2.57b)$$

or

$$RQ(I) = \int_{B_{ne}} \left(N_i \mathbf{n} \cdot \mathbf{K} \cdot \frac{\rho}{\rho_w} \nabla z - N_i q_n \right) dB \qquad (2.57c)$$

where q_v, q_c, and q_n are the rainfall/evaporation, Cauchy, or Neumann fluxes.

Subroutine BANSOL

This subroutine is called by the subroutine VELT and HMCHYD to solve for the matrix equation of the type:

$$[C]\{x\} = \{y\} \qquad (2.58)$$

where [C] is the coefficient matrix and $\{x\}$ and $\{y\}$ are two vectors. $\{x\}$ is the unknown to be solved and $\{y\}$ is the known load vector. The computer returns the solution $\{y\}$ and stores it in $\{y\}$. The computation is a standard banded Gaussian direct elimination procedure.

Subroutine PISS

This subroutine is called by either HMCHYD or HMCTRN which is determined by identification integer, IFT, to solve for the matrix equation of the type

$$[C]\{x\} = \{y\} \qquad (2.59)$$

where [C] is the coefficient matrix and $\{x\}$ and $\{y\}$ are two vectors. $\{x\}$ is the unknown to be solved, and $\{y\}$ is the known load vector. The computer returns the solution $\{x\}$ and stores it in $\{x\}$. The computation is a pointwise iteration method. Three options are provided for the point iteration solution strategy, the under-relaxation, Gauss-Seidel, and over-relaxation.

Subroutine VELT

This subroutine calls FQ4D and FQ3D to evaluate the element matrices and the derivatives of the total head. It then sums over all element matrices to form a matrix equation governing the velocity components at all nodal points. To save computational time, the matrix is diagonalized by lumping. The velocity components can thus be solved point by point. The computed velocity field is then returned to HTMICH or HMCHYD through the argument. This velocity field is also passed to subroutine BCPREP to evaluate the Darcy flux across the seepage-infiltration-evapotranspiration surfaces, and to subroutine HMCTRN to perform the transport computation.

Subroutine FQ4D and FQ3D

Subroutine FQ4D and FQ3D are called by the subroutine VELT to compute the element matrices given by

$$QB(I,J) = \int_{B_e} N_i^e N_j^e dR \qquad (2.60a)$$

where N_i^e and N_j^e are the base functions for nodal point i and j of element e, respectively. Subroutine FQ4D and FQ3D also evaluate the element load vector:

37

$$QRX(I) = -\int_{R_e} N_i{}^e \mathbf{i} \cdot \mathbf{K_s} K_r \cdot \frac{\rho_w}{\rho} (\nabla N_j{}^e) h_j dR - \int_{R_e} N_i{}^e \mathbf{i} \cdot \mathbf{K_s} K_r \cdot \nabla z dR \qquad (2.60b)$$

$$QRZ(I) = -\int_{R_e} N_i{}^e \mathbf{k} \cdot \mathbf{K_s} K_r \cdot \frac{\rho_w}{\rho} (\nabla N_j{}^e) h_j dR - \int_{R_e} N_i{}^e \mathbf{k} \cdot \mathbf{K_s} K_r \cdot \nabla z dR \qquad (2.60c)$$

where

H_j = the total head at nodal point j,

$\mathbf{i}$ = the unit vector along the x-coordinate,

$\mathbf{k}$ = the unit vector along the z-coordinate,

$\mathbf{K_s}$ = the saturated hydraulic conductivity tensor,

K_r = the relative hydraulic conductivity.

FQ4D is used to evaluate the integration for quadrilateral elements and FQ3D for triangular elements.

Subroutine HMCTRN

The subroutine HMCTRN controls the entire sequence of transport computations. HMCTRN calls subroutine THNODE to compute the value of moisture content plus bulk density times distribution coefficient in the case of linear isotherm; subroutine DISPC to calculate the dispersion coefficient (Eq. (2.16)) associated with each Gaussian point in every element; subroutine AFABTA to obtain upstream weighting factors based on velocity and dispersivity; subroutine TASEMB to assemble the element matrices over all elements; subroutine TBC to implement the boundary conditions; subroutine TBC1 to apply intra-boundary conditions with the implementation of the slave point concept to circumvent the incompatibility problem; subroutine SOLVE or subroutine PISS to solve the resulting matrix equations with the direct Gaussian elimination method or pointwise iteration methods; subroutine FLUX to compute material flux; subroutine TSFLOW to calculate the fluxes through all types of boundaries and accumulated in the media; subroutine TPRINT to print out the results; subroutine GNTRAK to perform particle tracking of global nodes; subroutine HPTRAK to perform particle tracking of fine nodal points, and subroutine ADVBC to implement boundary conditions in the

38

Lagrangian step; subroutine SFDET to determine sharp front elements; subroutine FGDET to imbed subelements into every sharp front element; subroutine ISEHIL to prepare ISE array which stores the indices of subelements and to determine the activation of the points with the highest or lowest concentrations in each subelement; subroutine DFPREP to prepare the fine mesh nodes and elements for diffusion zooming. Figure 2.4 shows the flow chart of this subroutine.

Subroutine THNODE

This subroutine is called by HTMICH and HMCTRN to compute $(\theta + \rho_b K_d)$ for the linear isotherm model.

Subroutine DISPC

Subroutine DISPC calculates the dispersion coefficient associated with each Gaussian point in every element. The three components of the dispersion coefficient tensor **D** are calculated using Eq. (2.16) and stored in AKDC array.

Subroutine AFABTA

This subroutine calculates the values of the upstream weighting factors along 4 or 3 sides of all elements.

Subroutine TASEMB

This subroutine calls TQ4 and TQ3 to evaluate the element matrices. It then sums over all element matrices to form a global matrix equation governing the concentration distribution at all nodes.

Subroutine TQ4 and Subroutine TQ3

These subroutines are called by the subroutine TASEMB to compute the element matrix given by

$$QA(I,J) = \int_{R_e} N_i^e \theta N_j^e dR \ , \tag{2.61a}$$

Figure 2.4 Program structure of 2DFATMIC (Transport Part 1 of 2)
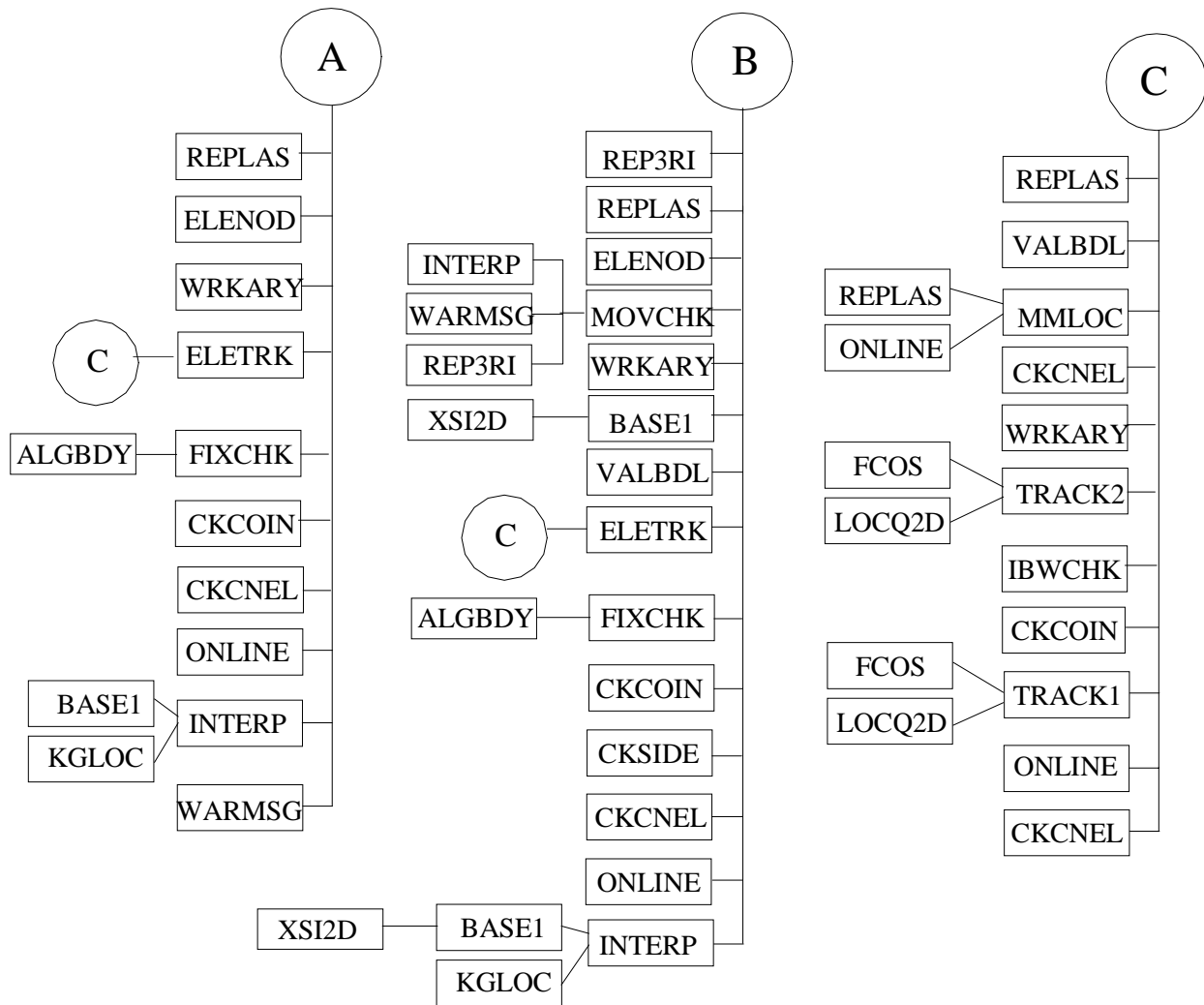
Figure 2.4 Program structure of 2DFATMIC (Transport Part 2 of 2)

$$QAA(I,J) = \int_{R_e} N_i^e \rho_b K_d N_j^e dR \ , \tag{2.61b}$$

$$QB(I,J) = \int_{R_e} (\nabla N_i^e) \cdot \theta \mathbf{D} \cdot (\nabla N_j^e) dR \ , \tag{2.61c}$$

$$QBB(I,J) \ = \ \int_{R_e} N_i^e \Lambda (\theta + \rho_b K_d) N_j^e dR \tag{2.61d}$$

$$QV(I,J) = \int_{R_e} N_i^e \mathbf{V} \cdot (\nabla N_j^e) dR \ , \tag{2.61e}$$

$$QC(I,J) = \int_{R_e} N_i^e (\frac{\rho^*}{\rho} Q - \frac{\rho_w}{\rho} \mathbf{V} \cdot \nabla \frac{\rho}{\rho_w}) N_j^e dR \ , \tag{2.61f}$$

$$QD(I,J) \ = \ \pm \int_{R_e} N_i \ (\theta + \rho_b K_d) \ f(C_1, C_2, C_3, C_s, C_o, C_n, C_p) \ N_j dR \tag{2.61g}$$

where $f(C_1, C_2, C_3, C_s, C_o, C_n, C_p)$ should be evaluated using those seven dissolved concentrations at previous iteration. Subroutines FQ3 and FQ4 also calculate the element load vector given by:

$$QR(I) = \int_{R_e} N_i^e QC_{in} dR \ , \tag{2.61h}$$

Subroutine SHAPE

This subroutine is called by subroutines TQ4D, TQ4 and Q4R to evaluate the value of the base and weighting functions and their derivatives at a Gaussian point. The computation is straightforward.

Subroutine BASE1

This subroutine is called by INTERP, INTERPK, ADVRX, SFDET, TASEMB, and HMCTRN to compute the base function values associated with a specified point based on the given two-dimensional global coordinates. For the cases of quadrilateral elements, it calls XSI2D to calculate the local coordinates, and

42

computes base functions with these determined local coordinates. For the cases of triangular elements, the base functions can be analytically determined based on the given global coordinates.

Subroutine XSI2D

This subroutine is called by BASE1 to compute the local coordinate of an element given the global coordinate within that element. With the local coordinate, the base functions can then easily be computed by formula.

Subroutine TBC

This subroutine incorporates Dirichlet, variable boundary, Cauchy, and Neumann boundary conditions. For a Dirichlet boundary condition, an identity is generated for each Dirichlet nodal point. Any other equations having this nodal variable are modified accordingly to simplify the computation. For a variable surface, the integration of the normal velocity times the incoming concentration is added to the load vector and the integration of normal velocity is added to the matrix. For the Cauchy boundaries, the integration of Cauchy flux is added to the load vector and the integration of normal velocity is added to the matrix. For the Neumann boundary, the integration of gradient flux is added to the load vector.

Subroutine Q2CNVB

These subroutines are called by the subroutines TBC to compute the surface node flux of the type

$$RQ(I) = \int_{R_e} N_i^e q dB \ ,$$

(2.62a)

where q is either the Cauchy flux, Neumann flux, or $\mathbf{n \cdot V}C_v$. It also computes the boundary element matrices

$$BQ(I,J) = \int_{R_e} N_i^e \mathbf{V} N_j^e dR \ ,$$

(2.62b)

43

Subroutine TBC1

This subroutine is called whenever the total number of nodes for composing the matrix is greater than the total number of global nodes, i.e., the diffusion zooming scheme is employed. The "slave point" concept takes care of the incompatibility for the intraboundary points between rough and smooth regions. This subroutine implements the concept so that the entries for the intraboundary points of the matrix equation can be modified. If there are diffusion fine grids falling on the global boundaries, the "slave point" concept also resolves the problems of implementation of boundary conditions for these fine grids. Subroutine SOLVE

This subroutine is called by HMCTRN to solve a matrix equation of the type

$$[C]\{x\} = \{y\} \tag{2.63}$$

where [C] is the coefficient matrix and $\{x\}$ and $\{y\}$ are two vectors. $\{x\}$ is the unknown to be solved and $\{y\}$ is the known load vector. The computer returns the solution $\{y\}$ and stores it in $\{y\}$. SOLVE uses a standard banded Gaussian direct-elimination procedure.

Subroutine FLUX

This subroutine calls TQ3D and TQ4D to evaluate the element matrices and the derivatives of concentrations. It then sums over all element matrices to form a matrix equation governing the flux components at all nodal points. To save computational time, the matrix is diagonalized by lumping. The flux components due to dispersion can thus be solved point by point. The flux due to the velocity is then added to the computed flux due to dispersion. The computed total flux field is then returned to HTMICH and HMCTRN through the argument.

Subroutine TQ4D and TQ3D

Subroutine TQ4D and TQ3D are called by the subroutine FLUX to compute the element matrices given by

$$QB(I,J) = \int\limits_{R_e} N_i^{\,e} N_j^{\,e} dR \quad , \tag{2.64a}$$

where $N_i^{\,e}$ and $N_j^{\,e}$ are the basis functions for nodal point i and j of element e, respectively which are computed by subroutine SHAPE in TQ4D. Subroutine TQ4D also evaluates the element load vector:

$$QRX(I) = - \int\limits_{R_e} N_i^{\,e} \mathbf{i} \cdot \theta \mathbf{D} \cdot (\nabla N_j^{\,e}) C_j dR \quad , \tag{2.64b}$$

$$QRZ(I) = - \int\limits_{R_e} N_i^{\,e} \mathbf{k} \cdot \theta \mathbf{D} \cdot (\nabla N_j^{\,e}) C_j dR \quad , \tag{2.64c}$$

where C_j is the concentration at nodal point j, $\mathbf{i}$ is the unit vector along the x-direction, $\mathbf{k}$ is the unit vector along the z-coordinate, θ is the moisture content, and $\mathbf{D}$ is the dispersion coefficient tensor. Subroutine TQ4D is used for quadrilateral elements and TQ3D is used for triangular elements.

Subroutine TSFLOW

This subroutine is used to compute the fluxes through various types of boundaries and the accumulation rate of material in the region of interest. FRATE(7) is to store the flux through the whole boundary

$$FRATE(7) = \int\limits_{B} (F_x n_x + F_z n_z) dB \quad , \tag{2.65}$$

where B is the global boundary of the region of interest; F_x, and F_z are the flux components; and n_x, and n_z are the directional cosines of the outward unit vector normal to the boundary B. FRATE(1) stores the fluxes through Dirichlet boundary B_d. FRATE(2) and FRATE(3) store the fluxes through Cauchy and Neumann

boundaries, respectively. FRATE(4) and FRATE(5) store incoming flux and outgoing flux rates, respectively, through the variable boundaries B_v^- and B_v^+, as given by

$$FRATE(1) = \int_{B_d} (F_x n_x + F_z n_z) dB \quad , \tag{2.66a}$$

$$FRATE(2) = \int_{B_c} (F_x n_x + F_z n_z) dB \quad , \tag{2.66b}$$

$$FRATE(3) = \int_{B_n} (F_x n_x + F_z n_z) dB \quad , \tag{2.66c}$$

$$FRATE(4) = \int_{B_v^-} (F_x n_x + F_z n_z) dB \quad , \tag{2.66d}$$

$$FRATE(5) = \int_{B_v^+} (F_x n_x + F_z n_z) dB \quad , \tag{2.66e}$$

where B_v^- and B_v^+ are that part of variable boundary where the fluxes are directed into the region and out from the region, respectively. FRATE(6) stores the fluxes through unspecified boundaries as

$$FRATE(6) = FRATE(7) - \sum_{I=1}^{5} FRATE(I) \tag{2.67}$$

FRATE(8) and FRATE(9) store the accumulation rates in dissolved and adsorbed phases, respectively, as given by

$$FRATE(8) = \int_R \frac{\partial \theta C}{\partial t} dR \quad , \tag{2.68}$$

$$FRATE(9) = \int_R \frac{\partial \rho_b S}{\partial t} dR \quad , \tag{2.69}$$

46

FRATE(10) stores the loss rate due to decay and FRATE(11) through FRATE(13) are set to zero as given by

$$FRATE(10) = \int_R \lambda(\theta C + \rho_b S)dR \ , \tag{2.70}$$

$$FRATE(11) = FRATE(12) = FRATE(13) = 0 \ , \tag{2.71}$$

FRATE(14) is used to store the source/sink rate as

$$FRATE(14) = \int_R [QC_{in}(1 + sign(Q)) + QC(1 - sign(Q))]/2dR \ , \tag{2.72}$$

If there is no numerical error in the computation, the following equation should be satisfied:

$$\sum_{I=7}^{14} FRATE(I) = 0 \tag{2.73}$$

and FRATE(6) should be equal to zero. The calculation of the contributions to FRATE(8), FRATE(9), FRATE(11), and FRATE(14) are in subroutine Q3R or Q4R.

Subroutine Q4R or Q3R

These subroutines are used to compute the contributions to FRATE(8), FRATE(9), FRATE(11), and FRATE(14) by quadrilateral and triangular elements, respectively:

$$QRM = \int_R \theta CdR \ , \tag{2.74a}$$

$$QDM = \int_R SdR \ , \tag{2.74b}$$

$$SOSM = \int_R [QC_{in}(1 + sign(Q)) + QC(1 - sign(Q))]/2dR \ , \tag{2.74c}$$

The computation of the above integrals is straightforward.

<u>Subroutine ADVBC</u>

This subroutine is called by HMCTRN to implement the boundary conditions. For Dirichlet boundaries, the Lagrangian concentration is specified. For variable boundaries, if the flow is directed out of the region, the fictitious particle associated with the boundary node must come from the interior nodes. Hence the Lagrangian concentration for the boundary node has already been computed from subroutine GNTRAK and the implementation for such a boundary segment is bypassed. Thus, care should be taken to ensure that the subroutine GNTRAK is called before the subroutine ADVBC. If the flow is directed into the region, the concentration of incoming fluid is specified for variable boundaries. An intermediate Lagrangian concentration is calculated according to

$$C_i^{**} = \int_{B_v^-} N_i V_n C_{in} dB / \int_{B_v^-} N_i V_n dB \ , \tag{2.75a}$$

where C_i^{**} is the concentration due to the boundary source at the boundary node i, V_n is the normal vertically integrated Darcy's velocity, and C_{in} is the concentration of incoming fluid.

Cauchy boundary conditions are normally applied to the boundary where flow is directed into the region, and the material flux of incoming fluid is specified. The intermediate Lagrangian concentration is thus calculated according to

$$C_i^{**} = \int_{B_c} N_i q_c dB / \int_{B_c} N_i V_n dB \ , \tag{2.75b}$$

where C_i^* is the concentration due to boundary source at the boundary node i, V_n is the normal component of the Darcy's velocity, and q_c is the Cauchy flux of the incoming fluid.

The final Lagrangian concentration on either the variable or Cauchy boundary is obtained by using the value C_i^{**} and C_i^n, which is the concentration at the previous time step, as follows.

48

$$C_i^* = \frac{\int_B N_i \theta N_j dB\, C_j^{**} + \int_B N_i \rho_b K_d N_j dB\, C_j^n}{\int_B N_i(\theta + \rho_b K_d)dB}$$

(2.75c)

Subroutine GNTRAK

This subroutine is called by HMCTRN to control the process of backward or forward particle tracking starting from global nodes. It is designed to get the Lagrangian concentrations of all the fictitious particles arriving at the global nodes at the current time step. In the subroutine, each particle is tracked one element by one element until either the tracking time is completely consumed or the particle encounters a specified boundary side. During the particle tracking, this subroutine calls ELETRK to track a particle in the element being considered. In order to make the particle tracking complete and remedy the given velocity field error on the unspecified boundaries, subroutine FIXCHK calls subroutine ALGBDY to continue tracking particles along the unspecified/Neumann boundaries. At the end of each particle tracking, this subroutine calls INTERP to calculate base functions such that the Lagrangian concentration can be computed by interpolation with those base function values.

Subroutine HPTRAK

This subroutine is called by HMCTRN to compute the locations and concentrations of all forward-tracked fine-mesh nodes. Basically, the algorithm of this subroutine is the same as that of subroutine GNTRAK.

Subroutine ADVRX

This subroutine is used to the following seven nonlinear simultaneous ordinary differential equations

49

$$\left(\theta + \rho_b K_{ds}\right)\frac{DC_s}{Dt} \;=\; -\left[\left(\theta + \rho_b K_{d1}\right)C_1\right]\left\{\frac{\mu_o^{(1)}}{Y_o^{(1)}}\left[\frac{C_s}{K_{so}^{(1)}+C_s}\right]\left[\frac{C_o}{K_o^{(1)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(1)}+C_p}\right]\right\}$$

$$-\left[\left(\theta + \rho_b K_{d2}\right)C_2\right]\left\{\frac{\mu_n^{(2)}}{Y_n^{(2)}}\left[\frac{C_s}{K_{sn}^{(2)}+C_s}\right]\left[\frac{C_n}{K_n^{(2)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(2)}+C_p}\right]\right\}$$

$$\tag{2.76a}$$

$$-\left[\left(\theta + \rho_b K_{d3}\right)C_3\right]\left\{\begin{array}{c}\dfrac{\mu_o^{(3)}}{Y_o^{(3)}}\left[\dfrac{C_s}{K_{so}^{(3)}+C_s}\right]\left[\dfrac{C_o}{K_o^{(3)}+C_o}\right]\left[\dfrac{C_p}{K_{po}^{(3)}+C_p}\right] \\[4mm] +\dfrac{\mu_n^{(3)}}{Y_n^{(3)}}\left[\dfrac{C_s}{K_{sn}^{(3)}+C_s}\right]\left[\dfrac{C_n}{K_n^{(3)}+C_n}\right]\left[\dfrac{C_p}{K_{pn}^{(3)}+C_p}\right]\mathbb{I}\left(C_o\right)\end{array}\right\}$$

$$\left(\theta + \rho_b K_{do}\right)\frac{DC_o}{Dt} \;=\; -\left[\left(\theta + \rho_b K_{d1}\right)C_1\right]$$

$$\left\{\gamma_o^{(1)}\mu_o^{(1)}\left[\frac{C_s}{K_{so}^{(1)}+C_s}\right]\left[\frac{C_o}{K_o^{(1)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(1)}+C_p}\right]+\alpha_o^{(1)}\lambda_o^{(1)}\left[\frac{C_o}{\Gamma_o^{(1)}+C_o}\right]\right\} \tag{2.76b}$$

$$-\left[\left(\theta + \rho_b K_{d3}\right)C_3\right]\left\{\gamma_o^{(3)}\mu_o^{(3)}\left[\frac{C_s}{K_{so}^{(3)}+C_s}\right]\left[\frac{C_o}{K_o^{(3)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(3)}+C_p}\right]+\alpha_o^{(3)}\lambda_o^{(3)}\left[\frac{C_o}{\Gamma_o^{(3)}+C_o}\right]\right\}$$

$$\left(\theta + \rho_b K_{dn}\right)\frac{D_n}{Dt} \;=\; -\left[\left(\theta + \rho_b K_{d2}\right)C_2\right]$$

$$\left\{\gamma_n^{(2)}\mu_n^{(2)}\left[\frac{C_s}{K_{sn}^{(2)}+C_s}\right]\left[\frac{C_n}{K_n^{(2)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(2)}+C_p}\right]+\alpha_n^{(2)}\lambda_n^{(2)}\left[\frac{C_n}{\Gamma_n^{(2)}+C_n}\right]\right\} \tag{2.76c}$$

$$-\left[\left(\theta + \rho_b K_{d3}\right)C_3\right]\left\{\gamma_n^{(3)}\mu_n^{(3)}\left[\frac{C_s}{K_{sn}^{(3)}+C_s}\right]\left[\frac{C_n}{K_n^{(3)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(3)}+C_p}\right]+\alpha_n^{(3)}\lambda_n^{(3)}\left[\frac{C_n}{\Gamma_n^{(3)}+C_n}\right]\right\}$$

$$\left(\theta + \rho_b K_{dp}\right)\frac{DC_p}{Dt} = -\left[\left(\theta + \rho_b K_{d1}\right)C_1\right]\left\{\epsilon_o^{(1)}\frac{\mu_o^{(1)}}{Y_o^{(1)}}\left[\frac{C_s}{K_{so}^{(1)}+C_s}\right]\left[\frac{C_o}{K_o^{(1)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(1)}+C_p}\right]\right\}$$

$$-\left[\left(\theta + \rho_b K_{d2}\right)C_2\right]\left\{\epsilon_n^{(2)}\frac{\mu_n^{(2)}}{Y_n^{(2)}}\left[\frac{C_s}{K_{sn}^{(2)}+C_s}\right]\left[\frac{C_n}{K_n^{(2)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(2)}+C_p}\right]\right\}$$

$$(2.76d)$$

$$-\left[\left(\theta + \rho_b K_{d3}\right)C_3\right]\left\{\begin{array}{c}\epsilon_o^{(3)}\dfrac{\mu_o^{(3)}}{Y_o^{(3)}}\left[\dfrac{C_s}{K_{so}^{(3)}+C_s}\right]\left[\dfrac{C_o}{K_o^{(3)}+C_o}\right]\left[\dfrac{C_p}{K_{po}^{(3)}+C_p}\right] \\[2em] +\epsilon_n^{(3)}\dfrac{\mu_n^{(3)}}{Y_n^{(3)}}\left[\dfrac{C_s}{K_{sn}^{(3)}+C_s}\right]\left[\dfrac{C_n}{K_n^{(3)}+C_n}\right]\left[\dfrac{C_p}{K_{pn}^{(3)}+C_p}\right]I(C_o)\end{array}\right\}$$

$$\left(\theta + \rho_b K_{d1}\right)\frac{DC_1}{Dt} = \left(\theta + \rho_b K_{d1}\right)C_1\left\{\mu_o^{(1)}\left[\frac{C_s}{K_{so}^{(1)}+C_s}\right]\left[\frac{C_o}{K_o^{(1)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(1)}+C_p}\right]-\lambda_o^{(1)}\right\} \qquad (2.76e)$$

$$\left(\theta + \rho_b K_{d2}\right)\frac{DC_2}{Dt} = \left(\theta + \rho_b K_{d2}\right)C_2\left\{\mu_n^{(2)}\left[\frac{C_s}{K_{sn}^{(2)}+C_s}\right]\left[\frac{C_n}{K_n^{(2)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(2)}+C_p}\right]-\lambda_n^{(2)}\right\} \qquad (2.76f)$$

$$\left(\theta + \rho_b K_{d3}\right)\frac{DC_3}{Dt} = \left(\theta + \rho_b K_{d3}\right)C_3\left\{\mu_o^{(3)}\left[\frac{C_s}{K_{so}^{(3)}+C_s}\right]\left[\frac{C_o}{K_o^{(3)}+C_o}\right]\left[\frac{C_p}{K_{po}^{(3)}+C_p}\right]-\lambda_o^{(3)}\right.$$

$$(2.76g)$$

$$\left. +\mu_n^{(3)}\left[\frac{C_s}{K_{sn}^{(3)}+C_s}\right]\left[\frac{C_n}{K_n^{(3)}+C_n}\right]\left[\frac{C_p}{K_{pn}^{(3)}+C_p}\right]I(C_o) - \lambda_n^{(3)}I(C_o)\right\}$$

This subroutine is called right after the Lagrangian concentrations have been obtained.

Subroutine RXRATE

This subroutine is called by subroutine ADVRX, and TASEMB during steady state simulations. Basically the subroutine calculates the removal rates of 7 chemical components which are represented as the terms within the braces on the right hand side of Eqs. (2.76a) through (2.76g). The values of each bracket within the braces are returned to the calling subroutines for each component.

Subroutine SFDET

This subroutine determines if an element is a rough element based on the prescribed error tolerance criteria shown in Eq.(2.46). If the M-th element is a rough element, the array $IE(M,7)$ is activated to 1.

Subroutine FGDET

This subroutine generates regular fine grids within each rough element based on the information of $IE(M,7)$ resulting from subroutine SFDET. It calls subroutine HPTRAK to obtain the Lagrangian concentrations of each activated fine grid.

Subroutine ISEHIL

This subroutine removes all the forward-tracked nodes in a smooth element and stores the indices of subelements into ISE array. In addition to regular refinement, subroutine ISEHIL also captures all the highest and lowest concentrations within each subelement. The subelements, where the high-low points are located, are determined by subroutine KGLOC. Once these high-low points are activated, subroutine TRIANG is called to triangulate this subelement and the indices of each triangular are also stored in ISE array.

Subroutine TRIANG

This subroutine calls subroutine CENTER and GRID to triangulate all the activated nodes in each subelement according to the Delauney triangulation method.

Subroutine DFPREP

This subroutine prepares all the needed information for assembling the fine grid elemental matrices.

It calls subroutines INTERPK to obtain the Lagrangian concentrations at each fine nodal points of the diffusion step. This subroutine also prepares element indices for each subelement in the Eulerian step for composing the matrix equation and stores the arrays for the intra-boundary points between rough and smooth regions to circumvent the incompatibility by implementing the "slave point" concept.

Subroutine WARMSG

The arguments passed to this subroutine are N, MAXN, SUBNAM, VARNAM, and NO. The stop statement is activated whenever N is greater than MAXN, and a message is written in the output file to indicate which variable is overflow in subroutine SUBNAM.

Subroutine INTERPK

This subroutine has the same capability as subroutine INTERP. Moreover, this subroutine once called would interpolate the concentrations of all components instead of one component as in subroutine INTERP.

Subroutine KGLOC

This subroutine is called by subroutine INTERP, INTERPK, and ISEHIL to obtain the subelement on which the peak/valley point falls if the global element is a rough element.

Subroutine ELENOD

This subroutine determines the number of nodes of element M by using the IE(M,4) information.

Subroutine VALBDL

This subroutine calculates two interpolated values by the passed working arrays and basis functions.

Subroutine CENTER

This subroutine is called by TRIANG to calculate the geometric center of each triangle.

Subroutine GRID

This subroutine is called by subroutine TRIANG to triangulate all the nodes in each subelement according to the Delaunay triangulation method. This triangular information is used by GNTRAK for backward-tracked interpolated concentration of next time step.

Subroutine NEWTRI

This subroutine is called by subroutine GRID to produce new triangles in each subelement generated in rough global elements.

Subroutine CKTRI

This subroutine is called by subroutine GRID to check the existence of the triangles produced by subroutine NEWTRI; i.e., to keep the triangles or not.

Subroutine ONLINE

This subroutine adjusts the particle coordinates to be on the same line with the other two points.

Subroutine REPLAS

This subroutine replaces the last four arguments with the first four arguments.

Subroutine WRKARY

This subroutine prepares four real and one integer working arrays for later usage.

Subroutine REP3RI

This subroutine replaces the last three real and one integer arguments with the first four arguments.

Subroutine MOVCHK

This subroutine determines the concentrations and travel time of a fixed particle.

Subroutine ELETRK

This subroutine is called by GNTRAK and HPTRAK to implement particle tracking in one element. In order to increase the accuracy of particle tracking, regular refinement is considered to make the element be composed of as many subelements as wished, and the tracking is executed one subelement by one subelement. This subroutine is designed to control the process of particle tracking among the subelements. During the particle tracking, this subroutine calls (1) TRACK1 to track a particle in the subelement being considered if that particle is right standing on a node of the subelement, and (2) TRACK2 to track a particle if that particle

54

is not on any nodes of the subelement. The tracking will not be stopped until either the tracking time is completely consumed or the particle encounters a side of the element.

Subroutine FIXCHK

This is a control panel to check the ongoing process when a particle hits the boundary of the region of interest. The backward tracked concentrations are obtained by interpolation if the boundary is specified including Dirichlet, Cauchy, and Variable types. Otherwise, the particle tracking continues along the unspecified boundary till either the specified boundaries are encountered or tracking time is consumed.

Subroutine ALGBDY

This subroutine is called by FIXCHK to control the process of backward or forward particle tracking along the unspecified boundaries. In the subroutine, the particle tracking is executed one boundary side by one boundary side based on the nodal velocity component along the side being considered. The tracking will not be stopped until either the tracking time is completely consumed or the particle encounters a specified boundary side. For accuracy, using the average velocity of both the source point and the target point to locate the target point is firstly considered in the subroutine. However, if this average velocity approach is not able to deal with very complex velocity fields, the single velocity of the source point is used to determine the location of the target point.

Subroutine CKCNEL

This subroutine checks the elements connected to a specific side segment.

Subroutine CKCOIN

This subroutine checks to see if a specific point coincides with a global node.

Subroutine CKSIDE

This subroutine checks to see if a specific point is on a side segment.

Subroutine INTERP

This subroutine is called by GNTRAK and HPTRAK to obtain the Lagrangian concentrations by finite

element interpolation of the concentrations of the previous time step. A backward-tracked fictitious particle can either land in a smooth element or a rough element. If it lands in a rough element, a determination must be made as to which subelement of the rough element it would land? This determination is made by calling subroutine KGLOC.

Subroutine MMLOC

This subroutine is called by ELETRK to locate the particle associated with a specific subelement for subsequent elemental tracking. If this particle coincides with the nodes of a subelement, ICODE=0 is returned.

Subroutine TRACK1

This subroutine is called by ELETRK to compute the particle tracking in a specified subelement when the source point coincides with a global node of the element. This subroutine determines (1) whether the particle would move backwards or forwards into the element or not, (2) which side of the element the particle would head onto if the particle does move backwards or forwards into this element, and (3) where the target point would be. After determining onto which side the particle is going to move, this subroutine computes the exact location of the target point on the side analytically. For accuracy, using the average velocity of both the source point and the target point to locate the target point is considered first in the subroutine. However, if this average velocity approach is not able to deal with very complex velocity fields, the single velocity of the source point is used to determine the location of the target point.

Subroutine TRACK2

This subroutine is called by ELETRK to compute the particle tracking in a specified subelement when the source point does not coincide with a global node of the element. This subroutine determines (1) whether the particle would move backwards or forwards into the element, (2) which side of the element the particle would head onto if the particle does move backwards or forwards into this element, and (3) where the target point would be. After determining which side the particle is going to move onto, this subroutine computes the exact location of the target point on the side analytically. For accuracy, using the average velocity of both the

56

source point and the target point to locate the target point is considered first in the subroutine. However, if this average velocity approach is not able to deal with very complex velocity fields, the single velocity of the source point is used to determine the location of the target point.

Subroutine LOCQ2D

This subroutine locates the target point of a particle tracking on a line segment in a specified element. All the computations are made according to the average velocity approach and the single velocity approach, as the index parameter IJUDGE is 1 and 2, respectively. When the average velocity approach is considered, the Newton-Ralphson method is used to solve a nonlinear algebraic equation such that the local coordinates of the target point on the pre-determined element segment can be determined. With these local coordinates, the location of the target point can be easily determined based on both the velocity of the source point and the geometrical relationship between the source point and the pre-determined element side.

Function FCOS

This function computes the cross of the vector of a given line segment with a specified vector whose starting point stands on the line. The result helps to determine where the endpoint of the specified vector is located.

Subroutine IBWCHK

This subroutine is called whenever the "in-element tracking" hits the global element side to determine on which element side the target point is located.

3. ADAPTATION OF 2DFATMIC TO SITE SPECIFIC APPLICATIONS

The following describes what one should do to the 2DFATMIC code for each site-specific application and what data files should be prepared.

3.1 Control-integer Specifications

For each site-specific problem, the users only need to specify the size of the problem by assigning 68 maximum control-integers with the PARAMETER statement in the MAIN program. The list and definitions of the maximum control-integers required for both flow and transport simulations are given below:

Maximum Control-Integers for the Spatial Domain

    MAXNPK = maximum no. of nodal points;
    MAXELK = maximum no. of elements;
    MXBESK = maximum no. of boundary-element surfaces;
    MXBNPK = maximum no. of boundary nodal points;
    MAXBWK = the maximum number of global nodes and/or fine nodes connected to any nodal point
             if the matrix equation is solved by PISS or maximum no. of bandwidth which is defined
             as 2*(maximum difference of global nodal number in an element)+1 if the matrix
             equation is solved by the direct Gaussian elimination;
    MXJBDK = maximum no. of nodal points connected to any nodal point resulting from all global and
             all subelements surrounding that point;
    MXKBDK = maximum no. of elements connected to any nodal point;
    MXADNK = maximum no. of points used to solve matrix equations for transport;

Maximum Control-Integers for the Time Domain

    MXNTIK = maximum no. of time steps;
    MXNDTK = maximum no. of DELT changes;

Maximum Control-Integers for Material and Soil Properties

    MXMATK = maximum no. of material types;
    MXSPPK = maximum no. of soil parameters per material to describe soil characteristic curves;
    MXMPPK = maximum no. of material properties per material;

The maximum control-integers for flow simulations and their definitions are given as the following:

Maximum Control-Integers for Source/sinks, flow

    MXSELF = maximum no. of source elements;
    MXSPRF = maximum no. of source profiles;
    MXSDPF = maximum no. of data points on each element source/sink profile;
    MXWNPF = maximum no. of well nodal points;
    MXWPRF = maximum no. of well source/sink profiles;
    MXWDPF = maximum no. of data points on each well source/sink profile;

Maximum Control-Integers for Cauchy Boundary Conditions, flow

    MXCNPF = maximum no. of Cauchy nodal points;
    MXCESF = maximum no. of Cauchy element surfaces;
    MXCPRF = maximum no. of Cauchy-flux profiles;
    MXCDPF = maximum no. of data points on each Cauchy-flux profile;

Maximum Control-Integers for Neumann Boundary Conditions, flow

    MXNNPF = maximum no. of Neumann nodal points;
    MXNESF = maximum no. of Neumann element surfaces;
    MXNPRF = maximum no. of Neumann-flux profiles;
    MXNDPF = maximum no. of data points on each Neumann-flux profile;

Maximum Control-Integers for Rainfall-Seepage Boundary Conditions, flow

    MXRNPF = maximum no. of variable nodal points;
    MXRESF = maximum no. of variable element surfaces;
    MXRPRF = maximum no. of rainfall profiles;
    MXRDPF = maximum no. of data point on each rainfall profile;

Maximum Control-Integers for Dirichlet Boundary Conditions, flow

    MXDNPF = maximum no. of Dirichlet nodal points;
    MXDPRF = maximum no. of Dirichlet total head profiles;
    MXDDPF = maximum no. of data points on each Dirichlet profile;

The maximum control-integers for transport simulations and their definitions are given as:

Maximum Control-Integers for Source/sinks, transport

    MXSELT = maximum no. of source elements;
    MXSPRT = maximum no. of source profiles;
    MXSDPT = maximum no. of data points on each element source/sink profile;
    MXWNPT = maximum no. of well nodal points;
    MXWPRT = maximum no. of well source/sink profiles;

MXWDPT = maximum no. of data points on each well source/sink profile;

Maximum Control-Integers for Cauchy Boundary Conditions, transport

   MXCNPT = maximum no. of Cauchy nodal points;
   MXCEST = maximum no. of Cauchy element surfaces;
   MXCPRT = maximum no. of Cauchy-flux profiles;
   MXCDPT = maximum no. of data points on each Cauchy-flux profile;

Maximum Control-Integers for Neumann Boundary Conditions, transport

   MXNNPT = maximum no. of Neumann nodal points;
   MXNEST = maximum no. of Neumann element surfaces;
   MXNPRT = maximum no. of Neumann-flux profiles;
   MXNDPT = maximum no. of data points on each Neumann-flux profile;

Maximum Control-Integers for Flowin-Flowout Boundary Conditions, transport

   MXVNPT = maximum no. of variable nodal points;
   MXVEST = maximum no. of variable element surfaces;
   MXVPRT = maximum no. of rainfall profiles;
   MXVDPT = maximum no. of data point on each rainfall profile;

Maximum Control-Integers for Dirichlet Boundary Conditions, transport

   MXDNPT = maximum no. of Dirichlet nodal points;
   MXDPRT = maximum no. of Dirichlet total head profiles;
   MXDDPT = maximum no. of data points on each Dirichlet profile;

Control-Integers for Number of Components in the system

   MXNCCK = maximum no. of components in this system;

Maximum Control-Integers for Refined System

   MXKGLDK = maximum no. of subelements in the Eulerian step;
   MXLSVK = maximum no. of subelement sides located on the intra-boundaries between extended
            rough and smooth regions;
   MXMSVK = maximum no. of global element sides located on the intra-boundaries between extended
            rough and smooth regions;
   MXNDBK = maximum no. of diffusion fine nodal-points located on the global boundary;
   MXNEPK = maximum no. of all forward tracked nodal points in the region of interest when the exact
            peak capture and oscillation free (EPCOF) numerical scheme is used. When EPCOF is
            not used, set MXNEPK = 1;
   MXEPWK = maximum no. of forward tracked nodal points in any rough element when the exact
            peak capture and oscillation free (EPCOF) numerical scheme is used. When EPCOF is
            not used, set MXEPWK = 1;

MXNPWK = maximum no. of fine nodal-points in any global element for particle tracking;

MXELWK = maximum no. of subelement elements in any global element for particle tracking;

MXNPWS = maximum no. of fine nodal-points in any global element which surrounds point sources/sinks for obtaining more accurate Lagrangian concentrations with injection/extraction wells in the region of interest;

MXELWS = maximum no. of subelements in any global element which surrounds any point wih a source/sink for obtaining more accurate Lagrangian concentrations with injection/extraction wells in the region of interest.

MXNPFGK = maximum no. of forward tracked nodal points over the region of interest or maximum no. of fine nodal points plus peak/valley nodal points;

MXKGLK = maximum no. of subelements in the Lagrangian step;

For flow simulations only, we demonstrate how to specify the above maximum control-integers with PARAMETER statement in the MAIN in the following by an example.

Assume that a region of interest is discretized by 11×11 nodes and 10×10 elements (Fig. 3.1). In other words, the region is discretized with 11 nodes along the longitudinal or x-direction and 11 nodes along the vertical or z-direction. Since the region has a total of nodes 11×11 = 121, the maximum number of nodes is MAXNPK = 121. The total number of elements is 10×10 = 100; i.e, MAXELK = 100. For this simple discretization problem, the maximum number of nodes connecting to any of the 121 nodes in the region of interest is 9; i.e., MXJBDK = 9; and the maximum number of elements connecting to any of the 121 nodes is 4; i.e. MXKBDK = 4. There will be 10 element segments each on the bottom and top faces of the region and 10 element segments each on the left and right faces of the region. Thus, there will be a total of 40 element segments; i.e., MXBESK = 40 . Similarly, we can compute the boundary nodes to be 40, i.e., MXBNPK = 40. Assume that the direct Gaussian elimination method is used to solve the matrix equation. Then MAXBWK is set equal to the half bandwidth plus 1. For this discretization, the computed half band width plus 1 is: 2*(13-1)+1 = 25; thus, MAXBWK = 25. Since this is a flow problem, no local grid refinement is involved; thus MXADNK = MAXNPK+0.
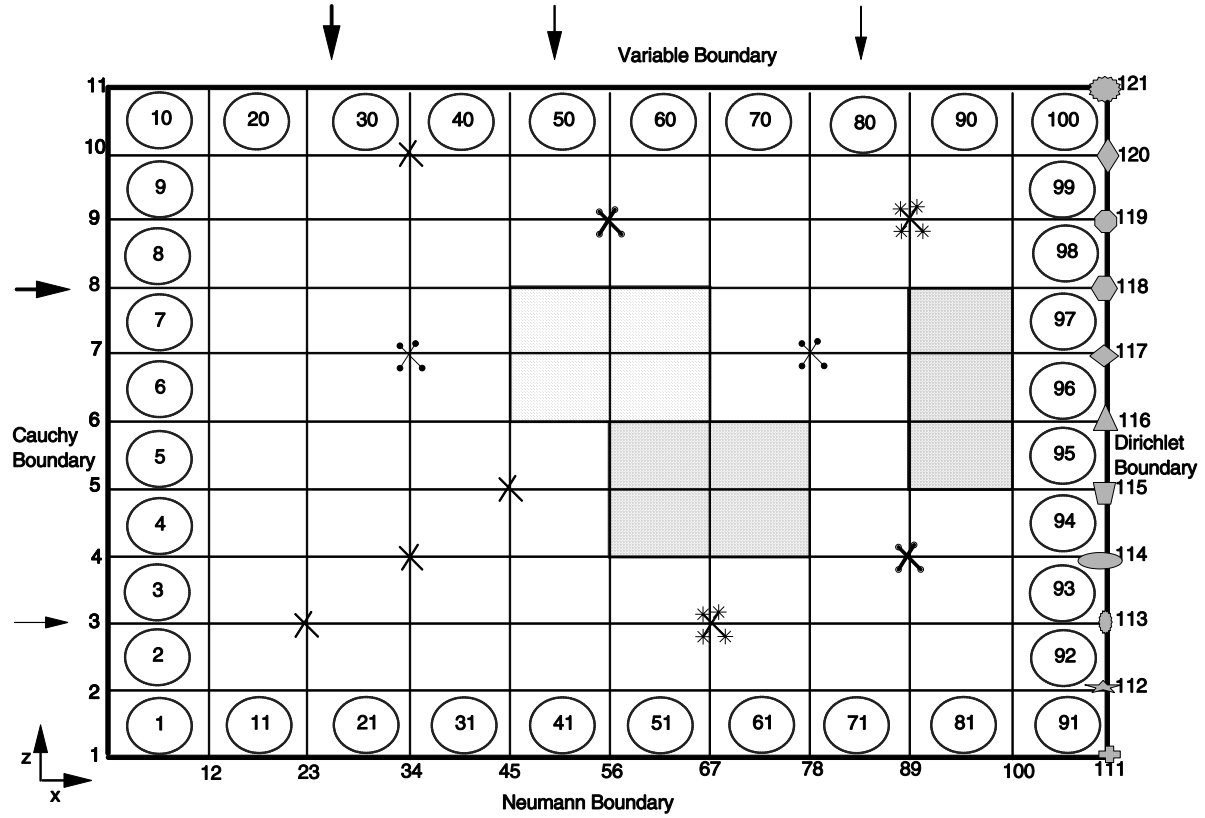
Figure 3.1 Finite Element Discretization and Source/Sink Layout of an Illustrated Example for Flow

In this model, six material properties (six saturated hydraulic conductivity components) per material were included for flow problems. Assume that the whole region of interest is made of three different kinds of materials. The characteristic curves of each material are assumed to be described by four parameters. We then have MXMATK = 3, MXMPPK = 6, and MXSPPK = 4. If we assume that we will make a 500 time step simulation and we will reinitiate the time step size for 20 times during our simulation, then we have MXNTIK = 500 and MXNDTK = 20.

We will assume that there will be a maximum of 11 elements that have the distributed sources/sinks (i.e., MXSELF = 11) and a maximum of 10 nodal points that can be considered as well sources/sinks (i.e., MXWNPF = 10). We will also assume that there will be two different distributed source/sink profiles and four distinct point source/sink profiles. Then we will have MXSPRF = 2 and MXWPRF = 4. Let us further assume that four data points are needed to describe the distributed source/sink profiles as a function of time and that 8 data points are required to describe point source/sink profiles (i.e., MXSDPF = 4 and MXWDPF = 8).

To specify maximum control-integers for the boundary conditions, let us assume that the top face is a variable boundary (i.e., on the air-soil interface, either ponding, infiltration, or evapotranspiration may take place). On the left face, fluxes from the adjacent aquifer are known. On the right face, the total head is assumed known. On the bottom face, natural drainage is assumed to occur (i.e., the gradient of the pressure head can be assumed zero). There are 11 nodes on the left face and 10 element segments; thus MXCNPF = 11 and MXCESF = 10. It is further assumed that there are two different fluxes going into the region through the left face and that each flux can be described by four data points as a function of time (i.e., MXCPRF = 2, and MXCDPF = 4). On the bottom surface, there are 11 nodes and 10 element segments. Since the gradient of pressure head on the bottom surface is zero, there is only one Neumann flux profile, and two data points, one at zero time and the other at infinite time, are sufficient to describe the constant value of zero. Hence, we have MXNNPF = 11, MXNESF = 10, MXNPRF = 1, and MXNDPF = 2. On the top face, there will be 11 nodes and 10 element segments. Let us assume that there are three different rainfall intensities that might be falling on the air-soil interface, and that each rainfall intensity is a function of time and can be described by 24 data points. With these descriptions, we have MXRNPF = 11, MXRESF = 10, MXRPRF = 3, and MXRDPF = 24. On the right face, there are 11 nodes. Let us assume that there are eleven different values of the total head, one each on a vertical line

of the right face.   We further assume that each of these twenty total heads can be described by 8 data

points as a function of time. We then have MXDNPF = 11, MXDPRF = 11, and MXDDPF = 8.

From the above discussion, the following PARAMETER statements can be used to specify the

maximum control-integers in the MAIN for the problem at hand:


```
PARAMETER(MAXELK=100,MAXNPK=121,MXBNPK=40,MXBESK=40,MAXBWK=25)
PARAMETER(MXJBDK=9,MXKBDK=4,MXADNK=MAXNPK+0)
PARAMETER(MXMATK=3,MXSPPK=4,MXMPPK=6)
PARAMETER(MXNTIK=500,MXNDTK=20)

PARAMETER(MXSELF=11,MXSPRF=2,MXSDPF=4,MXWNPF=10,MXWPRF=4,MXWDPF=8)
PARAMETER(MXCNPF=11,MXCESF=10,MXCPRF=2,MXCDPF=4)
PARAMETER(MXNNPF=11,MXNESF=10,MXNPRF=1,MXNDPF=2)
PARAMETER(MXRNPF=11,MXRESF=10,MXRPRF=3,MXRDPF=24)
PARAMETER(MXDNPF=11,MXDPRF=20,MXDDPF=8)

PARAMETER(MXSELT=1,MXSPRT=1,MXSDPT=1,MXWNPT=1,MXWPRT=1,MXWDPT=1)
PARAMETER(MXCNPT=1,MXCEST=1,MXCPRT=1,MXCDPT=1)
PARAMETER(MXNNPT=1,MXNEST=1,MXNPRT=1,MXNDPT=1)
PARAMETER(MXVNPT=1,MXVEST=1,MXVPRT=1,MXVDPT=1)
PARAMETER(MXDNPT=1,MXDPRT=1,MXDDPT=1)

PARAMETER(MXNCCK=1)

PARAMETER(MXKGLDK=1,MXLSVK=1,MXMSVK=1,MXNDBK=1)
PARAMETER(MXNEPK=1,MXEPWK=1)
PARAMETER(MXNPWK=1,MXELWK=1,MXNPWS=1,MXELWS=1)
PARAMETER(MXNPFGK=1,MXKGLK=1)
```


In the following, for transport simulations only, we demonstrate how to specify the maximum control-

integers with PARAMETER statements in the MAIN with an example.

Let us assume that a region of interest is discretized by 11×11 nodes and 10×10 elements (Fig. 3.2).

In other words, we are discretizing the region with 11 nodes along the longitudinal or x-direction and 11

nodes along the vertical or z-direction. Since we have a total of nodes $11 \times 11 = 121$, the maximum

number of nodes is <u>MAXNPK = 121</u>. The total number of elements is $10 \times 10 = 100$; i.e, <u>MAXELK = 100</u>. For this simple discretization problem, the maximum number of nodes connecting to any of the 121 nodes in the region of interest is 17; i.e., <u>MXJBDK = 17</u>. (e.g., there are four global elements connected to global node 25, thus there are 9 global nodes connected to node 25. In addition, there are four subelements surrounding node 25, thus 8 additional fine nodal points are connected to node 25. Hence the total number of global nodes and fine nodal points connected to node 25 is 17.) The maximum number of elements connecting to any of the 121 nodes is 4; i.e. <u>MXKBDK = 4</u>. There will be 10 element segments each on the bottom and top faces of the region and 10 element segments each on the left and right faces of the region. Thus, there will be a total of 40 element segments; i.e., <u>MXBESK = 40</u>. Similarly, the number of boundary nodes is 40; thus <u>MXBNPK = 40</u> (Fig. 3.2).
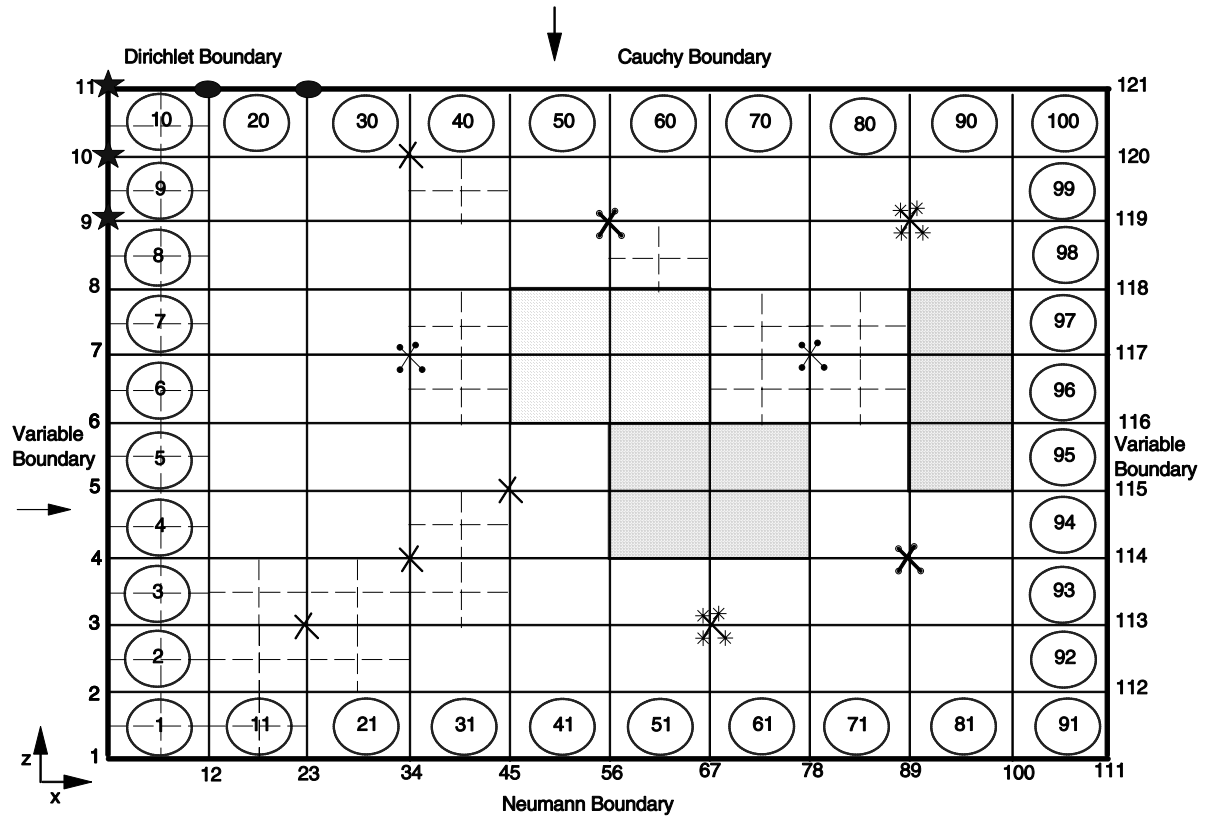
Figure 3.2 Finite Element Discretization and Source/Sink Layout of an Illustrated Example for Transport

We assume that there are 25 rough elements as shown in Figure 3.2. Each rough element is refined by 4 (2 x 2 = 4) subelements in the diffusion step computation. It is seen (Fig. 3.2) that the maximum number of nodal points (including both global and fine nodal points) connecting to any nodal points is 9. We assume that adaptive zooming will be enforced in the diffusion step of solving the transport equation, which necessitates the use of point iteration method to solve the matrix equation. Hence, MAXBWK is set to be the maximum number of nodal points connecting to any node; i.e. <u>MAXBWK = 9.</u> This is different from the flow case described previously, where the direct Gaussian elimination method was used to solve the matrix equation; thus MAXBWK was set to be the maximum half bandwidth plus 1.

66

Five additional diffusion points for each rough element will be added to the region of interest with a 2 x 2 subelement refinement. When some of the rough elements are connected, some of these diffusion points will be duplicated. For example, there are 101 additional diffusion points, not 5 X 25 = 125 points because some of the rough elements are connected. Thus, <u>MXADNK = MAXNPK + 101</u>. If all 25 rough elements were disconnected, MXADNK would be specified as <u>MXADNK = MAXNPK + 5 x 25 = MAXNPK + 125.</u> For sake of safety, sometime we may want to assign the latter value.

In this model, we have included eight material properties per material for transport problems. We will assume that the whole region of interest is composed of three different kinds of materials. We then have <u>MXMATK = 3</u> and <u>MXMPPK = 8</u>. If we assume that we will make a 500 time step simulation and we will reinitiate the change on the time step size for 20 times during our simulation, then we have <u>MXNTIK = 500</u> and <u>MXNDTK = 20</u>.

Because there are 25 rough elements and each rough element is refined by 2 x 2 = 4 subelements, the maximum number of subelements for assembling in the diffusion step is <u>MXKGLDK = 25 x 4 = 100.</u> Figure 3.2 shows that the number of global element sides located on the intra-boundaries between rough and smooth regions is 51, and the number of subelement sides on these intra-boundaries is 102. Hence, <u>MXMSVK = 51</u>, and <u>MXLSVK = 102</u>. The number of diffusion fine nodal points on the global boundary is <u>MXNDBK = 12</u> (Fig. 3.2). The values of MXMSVK naturally depends on where and how the rough elements are located in the region of interest. In the worst case when all 25 rough elements are disconnected, the MXMSVK would be equal to 25 x 4 = 100 because each rough element would be surrounded by four smooth elements. The value of MXLSVK would depend on how each rough element is refined. For this particular case, MXLSVK = 100 x 2 = 200 because every global element side is refined by 2 subelement sides.

We will assume that there will be a maximum of 11 elements that have the distributed sources/sinks (i.e., <u>MXSELT = 11</u>) and a maximum of 10 nodal points that can be considered as well sources/sinks (i.e., <u>MXWNPT = 10</u>) as shown in Figure 3.2. We will also assume that there will be two different distributed source/sink profiles and four distinct point source/sink profiles (Fig. 3.2). Then we will have <u>MXSPRT = 2</u> and <u>MXWPRT = 4</u>. Let us further assume that four data points are needed to describe the distributed source/sink profiles as a function of time and that 8 data points are required to describe point source/sink profiles (i.e., <u>MXSDPT = 4</u> and <u>MXWDPT = 8</u>).

To specify maximum control-integers for boundary conditions, let us assume that nodes 9, 10, 11, 22, and 33 (Fig. 3.2) are the Dirichlet boundary nodes. The remaining top face is a Cauchy boundary. The remaining left face and the whole right face are the variable boundaries (i.e., either Cauchy condition with known incoming concentration or Neumann condition with zero gradient of concentration). On the bottom face, the Neumann condition is assumed to occur with zero concentration gradient. For the five Dirichlet nodes, let us assume that there are two profiles to be applied to these five nodes. We further assume that each profile can be described by 8 data points for the Dirichlet boundary values. From this description, we have <u>MXDNPT = 5</u>, <u>MXDPRT = 2</u>, and <u>MXDDPT = 8.</u> There are 8 element segments which consist of 9 nodes on the remainder of the top face; thus <u>MXCNPT = 9</u> and <u>MXCEST = 8</u>. It is further assumed that there are two different fluxes going into the region through the top face and that each flux can be described by four data points as a function of time (i.e., <u>MXCPRT = 2</u>, and <u>MXCDPT = 4</u>). On the bottom face, there are 10 element segments which consist of 11 nodes. Since the gradient of concentration on the bottom surface is zero, there is only one Neumann flux profile, and two data points, one at zero time and the other at infinite time, are sufficient to describe the constant value of zero. Hence, we have <u>MXNNPT = 11</u>, <u>MXNEST = 10</u>, <u>MXNPRT = 1</u>, and <u>MXNDPT = 2</u>. On part of the left face and the whole right face, there are a total of 18 element segments, which consist of 20 nodal points. Let us assume that there are two concentration profiles imposed on the left face and one concentration

68

imposed on the right face, that is a total of 3 concentration profiles are used for the variable boundary condition. We further assume that each concentration profile is a function of time and can be described by 24 data points. With these descriptions, we have MXVNPT = 20, MXVEST = 18, MXVPRT = 3, and MXVDPT = 24.

There are seven components, say microbe #1, microbe #2, microbe #3, substrate, oxygen, nitrate, and nutrient, involved in this system; i.e., MXNCCK = 7.

The numerical schemes for solving transport equations are LEZOOMPC plus keeping EPCOF points in the Lagrangian step. For practical problems, EPCOF points will not be kept; thus, MXNEPK = 1, MXEPWK = 1. In the Lagrangian step, each element is assumed to be refined by 4 subelements (NXW = 2 and NZW = 2) for accurate tracking. With this assumption, we have MXNPWK = (NXW+1) x (NZW+1) = 9, MXELWK = (NXW x NZW) = 4. For each element connected to the sources/sinks, we assume to refine it with 3 x 3 elements for accurate computation of Lagrangian concentrations to yield MXNPWS = 16 and MXELWS = 9.

The specification of MXNPFGK and MXKGLK is much more involved. These two control integers depend on many things: (1) how all the nodal points (including global nodes and fine nodal points) at the beginning of a time-step simulation are forwardly tracked, (2) how many elements are rough at the end of the time-step computation, (3) how each rough element is refined, and (4) how many peak/valley points are kept. Figure 3.3 helps to illustrate how these two control integers are determined. Let us assume that at the beginning of a time-step computation, there are 24 global elements and 35 global nodes (Fig. 3.a). We further assume that four elements are rough and each rough element is refined by 2 x 2 = 4 subelements. The total number of nodes in the region of interest is 51: nodes 1 through 35 are global nodes and nodes 36 through 51 are fine nodal points. At this moment, the number of subelements would be 4 x 4 = *16*, and the number of fine nodal points would be 4 x (3 x 3) = **36** (in computing the number

of fine nodal points we have included the global nodes in rough elements) even though there are only 26

fine nodal points.
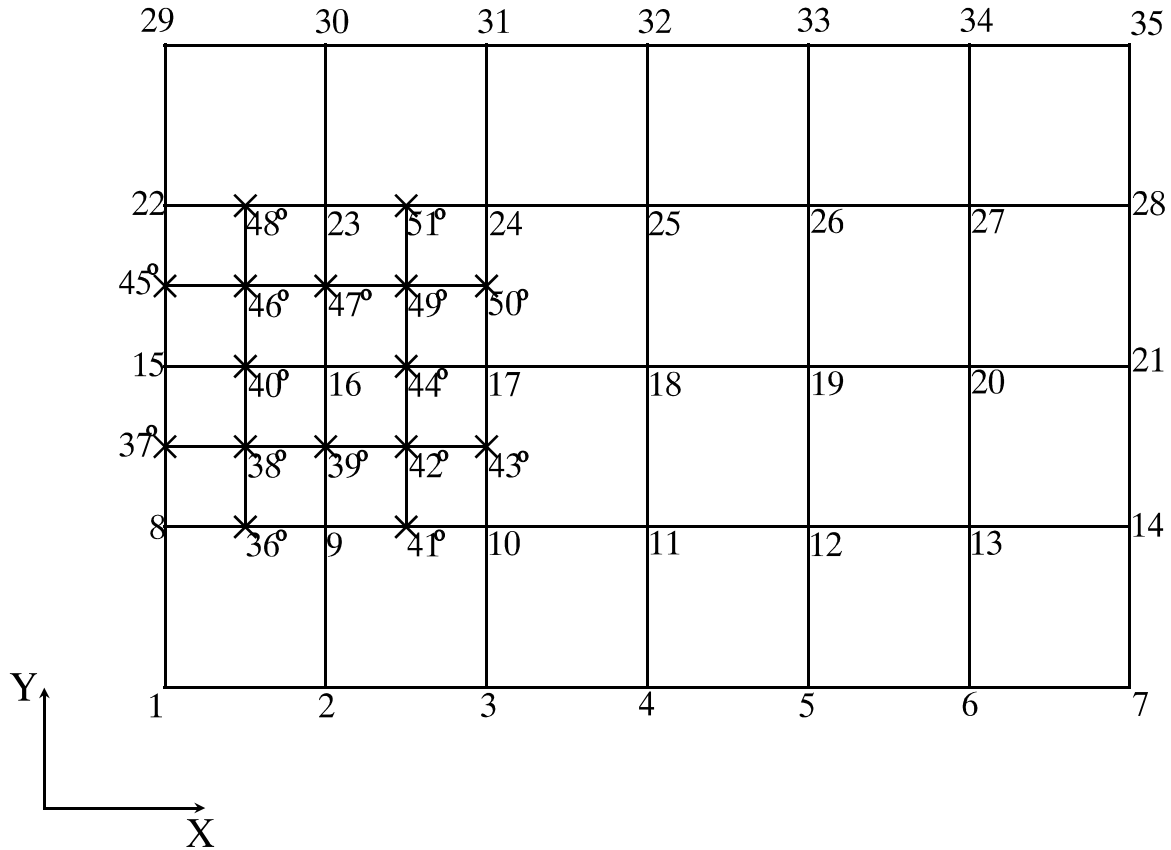
Figure 3.3a An Illustrated Example of Global Nodes and Fine Nodes for Initial Condition

According to the LEZOOMPC algorithm, we need to perform forward node tracking. Since we have

seven components, we need to perform a forward node tracking with respect to each component. Let us

assume, the forward tracking velocity with respect to Component 1 is given in Figure 3.3(b). We need

to forward track all 51 nodal points. After forward tracking, some of these 51 points will remain in the

region of interest and some of them will go out of the region. For this particular case, 32 nodal points

remain in the region of interest (Fig. 3.3b). These 32 nodal points are referred to as the intermediate-fine-

nodal points with respect to Component 1. Similarly, we have to perform forward tracking of 51 nodal

points with respect to Components 2 through 7.
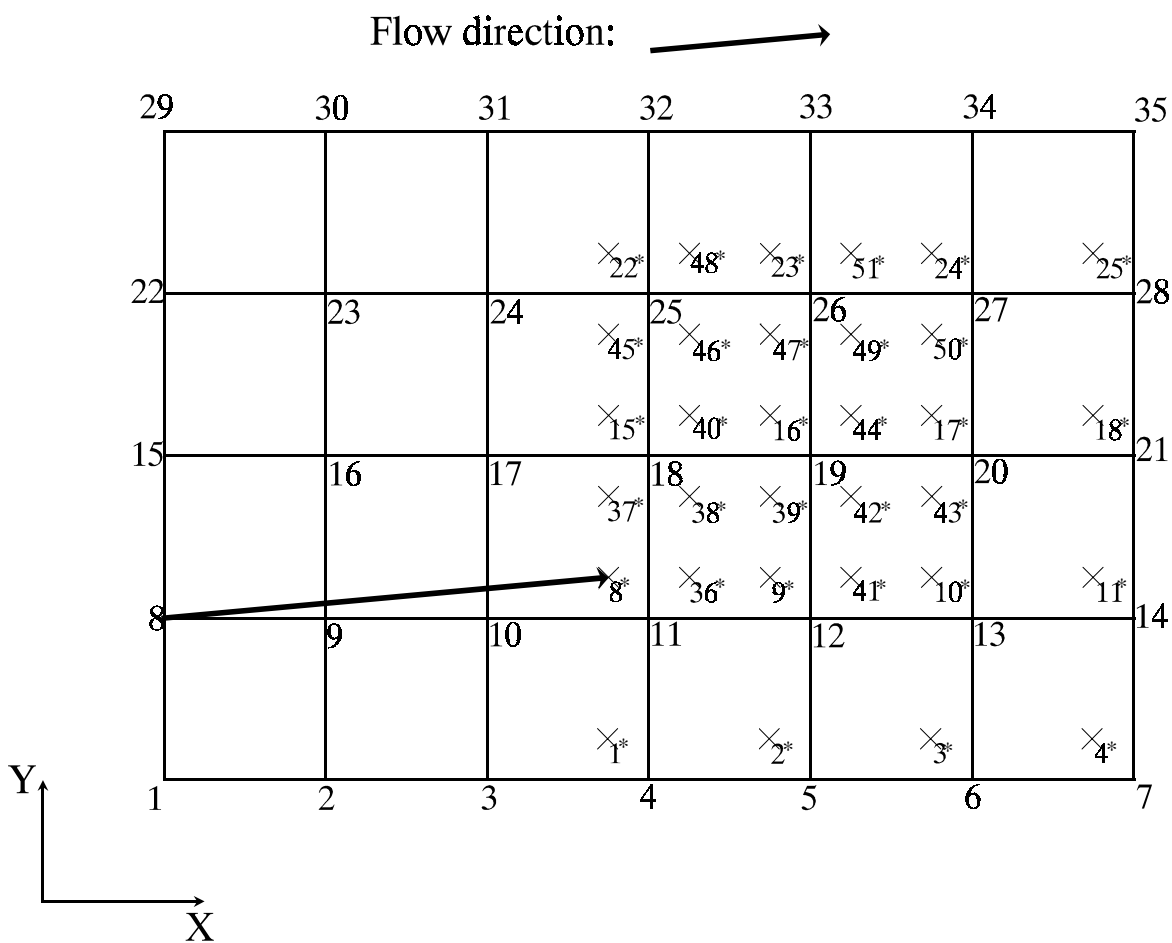
Figure 3.3b An Illustrated Example of Forward-tracked Fine Nodal Points

At the end of forward tracking, the number of fine nodal points would be the sum of all intermediate-

fine-nodal points that still remain in the region of interest.   In general, each chemical component has a

tracking velocity of its own; thus there will be a set of intermediate-fine-nodal points corresponding to

each component. Let us assume that the numbers of intermediate-fine-nodal points corresponding to Components 2 through 7 are 30, 34, 35, 36, 33, and 40, respectively. The number of fine nodal points would then be $32 + 30 + 34 + 35 + 36 + 33 + 40 =$ **240**.

The next step in LEZOOMPC algorithm is the determination of rough elements. Let's assume that we have determined 4 elements to be rough based on all the forward tracked nodes, and these four elements happen to be the elements formed by global nodes 11-12-19-18, 12-13-20-19, 18-19-26-25, and 19-20-27-26, respectively (Fig. 3.3c). The final step in the LEZOOMPC is the regular refinement and peak/valley capturing at the end of the time-step simulation. These four rough elements are refined with $4 \times (2 \times 2) = 16$ subelements and $4 \times (3 \times 3) = 36$ fine nodal points. Let us further assume that there is a peak/valley nodal point in each of 9 subelements out of the 16 subelements (Fig. 3.3c). The number of fine nodal points at the end of the time-step computation would be equal to $36 + 9 =$ **45** (Fig. 3.3c). For those 9 subelements that have peak/valley nodal points, each has to be triangularized. The number of triangles in each subelement depends on the number of peak/valley nodal points. The number of triangles can be computed with $2N_p + 2$ (where N_p is the number of peak/valley nodal points).   For this particular case, each subelement with a peak/valley nodal point can be triangularized with 4 triangular elements. Thus, the number of subelements after regular refinement and triangulation is equal to 9 (subelements with one peak/valley nodal points) x $4 + 7$ (subelements without peak/valley nodal points) $= 43$ (Fig. 3.3c).
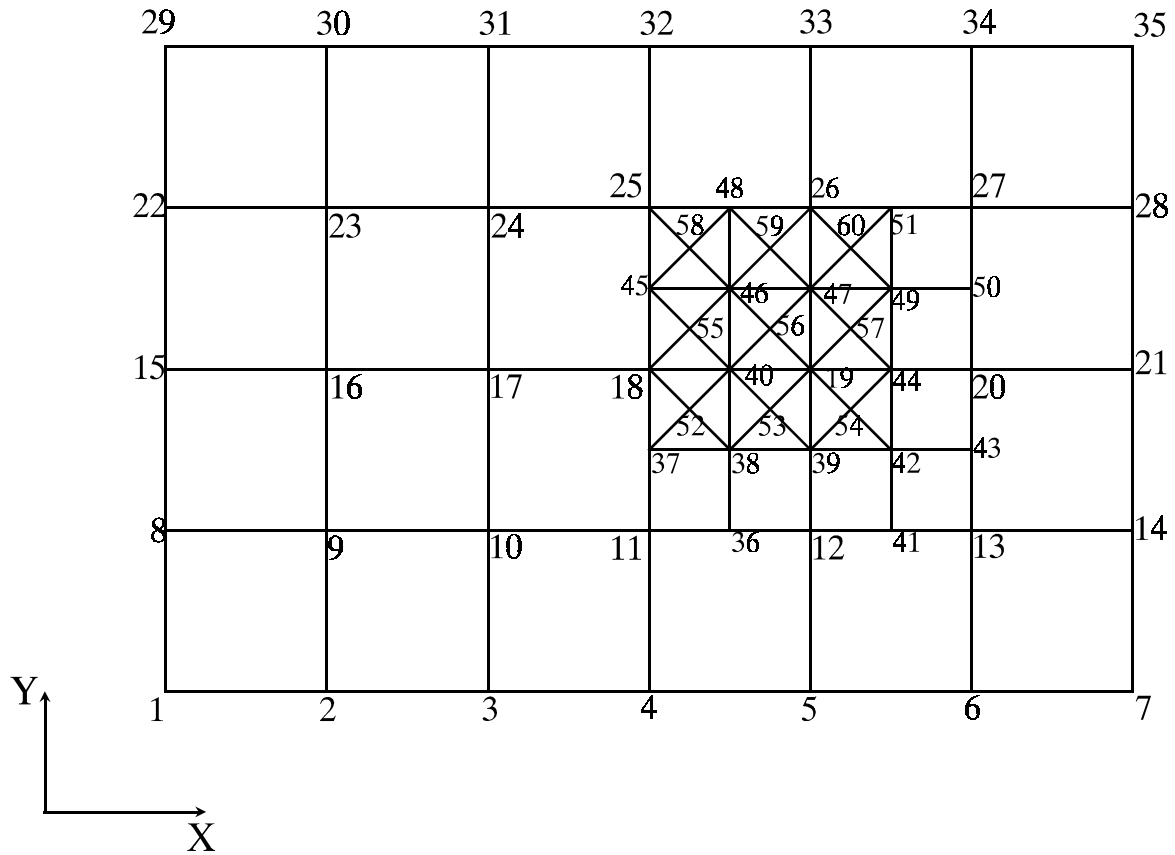
Figure 3.3c An Illustrated Example of Fine Nodal Points and Peak/Valley Points
at the End of the Time Step for Advection

The maximum number of fine-nodal points for the present time-step would be MXNPFGK = max (**36,240, 45**) = **240**. The maximum number of subelements would be MXKGLK = max (*16,43*) = *43*. The specification of MXNPFGK and MXKGLK should be large enough for all time steps, not just for a particular time step.

From the above discussion, the following PARAMETER statements can be used to specify the maximum control-integers in the MAIN for the problem at hand:

73

```
PARAMETER(MAXNPK=121,MAXELK=100,MXBNPK=40,MXBESK=40,MAXBWK=9)
PARAMETER(MXJBDK=9,MXKBDK=4,MXADNK=MAXNPK+125)
PARAMETER(MXMATK=3,MXSPPK=6,MXMPPK=8)
PARAMETER(MXNTIK=500,MXNDTK=20)

PARAMETER(MXSELF=1,MXSPRF=1,MXSDPF=1,MXWNPF=1,MXWPRF=1,MXWDPF=1)
PARAMETER(MXCNPF=1,MXCESF=1,MXCPRF=1,MXCDPF=1)
PARAMETER(MXNNPF=1,MXNESF=1,MXNPRF=1,MXNDPF=1)
PARAMETER(MXRNPF=1,MXRESF=1,MXRPRF=1,MXRDPF=1)
PARAMETER(MXDNPF=1,MXDPRF=1,MXDDPF=1)

PARAMETER(MXSELT=11,MXSPRT=2,MXSDPT=4,MXWNPT=10,MXWPRT=4,MXWDPT=8)
PARAMETER(MXCNPT=9,MXCEST=8,MXCPRT=2,MXCDPT=4)
PARAMETER(MXNNPT=11,MXNEST=10,MXNPRT=1,MXNDPT=2)
PARAMETER(MXVNPT=20,MXVEST=18,MXVPRT=3,MXVDPT=24)
PARAMETER(MXDNPT=5,MXDPRT=2,MXDDPT=8)

PARAMETER(MXNCCK=7)

PARAMETER(MXKGLDK=100,MXLSVK=102,MXMSVK=51,MXNDBK=12)
PARAMETER(MXNEPK=1,MXEPWK=1)
PARAMETER(MXNPWK=9,MXELWK=4,MXNPWS=16,MXELWS=9)
PARAMETER(MXNPFGK=240,MXKGLK=43)
```

3.2 Soil Property Function Specifications

Analytical functions are used to describe the relationships of water content, water capacity, and relative hydraulic conductivity with the pressure head. Therefore, the user must supply three functions to compute the water content, water capacity, and relative hydraulic conductivity based on the current value of the pressure head. The parameters needed to specify the functional form are read and stored in SPP. One example is shown in the subroutine SPROP in the source code. In this example, the water content, water capacity, and relative hydraulic conductivity are given by (van Genuchten 1980):

$$\theta \ = \ \theta_r \ + \ \frac{\theta_s - \theta_r}{[1 + (\alpha h)^n]^{\,m}} \tag{3.1}$$

$$\frac{d\theta}{dh} = \alpha(n-1)[1-f(\theta)]^m[f(\theta)](\theta_s-\theta_r) \tag{3.2}$$

$$K_r = [(\theta-\theta_s)/(\theta_s-\theta_r)]^{\frac{1}{2}}\{1-[1-f(\theta)]^m\}^2 \tag{3.3}$$

in which

$$f(\theta) = [\theta-\theta_r]/[\theta_s-\theta_r]^{1/m} \tag{3.4}$$

and

$$m = 1 - \frac{1}{n} \tag{3.5}$$

To further demonstrate how we should modify the subroutine SPROP in Appendix A to accommodate the material property functions that are different from those given by Eqs. (3.4) through (3.5), let us assume that the following Fermi types of functions are used to represent the unsaturated hydraulic properties (Yeh 1987):

$$\theta = \theta_r + (\theta_s-\theta_r)/\{1+\exp[-\alpha(h-h_\theta)]\} \tag{3.6}$$

$$d\theta/dh = \alpha(\theta_s-\theta_r)\exp[-\alpha(h-h_\theta)]/\{1+\exp[-\alpha(h-h_\theta)]\}^2 , \tag{3.7}$$

and

$$\log_{10}(K_r) = \epsilon/\{1+\exp[-\beta(h-h_k)]\} - \epsilon , \tag{3.8}$$

where θ_s, θ_r, α, and h_θ are the parameters for computing the water content and water capacity; and β, ϵ, and h_k are the parameters for computing the relative hydraulic conductivity.   The source code, subroutine SPFUNC, must be changed, for this example, to the following form for computing the moisture content and water capacity

```
WCR=SPP(1,MTYP,1)
WCS=SPP(2,MTYP,1)
ALPHA=SPP(3,MTYP,1)
HTHETA=SPP(4,MTYP,1)
EPS=SPP(1,MTYP,2)
```

```
      BETA=SPP(2,MTYP,2)
      HSUBK=SPP(3,MTYP,2)
C
C ------- SATURATED CONDITION
C
      IF(HNP.LE.0.0) THEN
       TH=WCS
       IF(ISP.EQ.1) GOTO 900
       DTH=0.0D0

       USKFCT=1.0D0
      ELSE
C
C ------ UNSATURATED CASE
C
      EXPAH=DEXP(-ALPHA*(HNP-HTHETA))
      TH=WCR+(WCS-WCR)/(1.0D0+EXPAH)
      IF(ISP.EQ.1) GOTO 900
      DTH=ALPHA*(WCS-WCR)*EXPAH/(1.0D0+EXPAH)**2
      AKRLOG=EPS/(1.0D0+DEXP(-BETA*(HNP-HSUBK))) -  EPS
      UKFCT=10.0D0**AKRLOG
C
```

3.3 Input and Output Devices

Five logical units are needed to execute 2DFATMIC. Units 15 and 16 are standard card input and line printer devices, respectively. Unit 11 must be specified to store the flow simulation results, which can be used for plotting purposes. Unit 12 must be specified to store the transport simulation results, which can be used for plotting purposes. Unit 13 is used to store the boundary arrays for later uses, if these arrays are computed for the present job. Unit 14 is used to store pointer arrays for later uses, if these arrays are generated for the present job. For large problems, our experience has indicated that it would take too much time to process the boundary arrays and to generate pointer arrays. Hence, it is advisable that for multijob executions, these boundary and pointer arrays should be computed only once and written on units 13 and 14, respectively. Once they are stored on units 13 and 14, the IGEOM described in Appendix A should be properly identified for the new job so they can be read via units 13 and 14, respectively.

76

4. SAMPLE PROBLEMS

To verify 2DFATMIC, three illustrative examples are used. The first one is a one-dimensional flow problem through a column. The second one is a one-dimensional transport problem through a column. The third one is a two-dimensional biodegradation problem.

4.1 Problem No. 1: One-Dimensional Column Flow Problem

Problem Description

This example is selected to represent the simulation of a one-dimensional problem with the revised FEMWATER. The column is 200 cm long and 50 cm wide (Figure 4.1). The column is assumed to contain the soil with a saturated hydraulic conductivity of 10 cm/day, a porosity of 0.45 and a field capacity of 0.1. The unsaturated characteristic hydraulic properties of the soil in the column are given as:

$$\theta = \theta_s - \left(\theta_s - \theta_r\right)\left(\frac{h - h_a}{h_b - h_a}\right) \tag{4.1}$$

and

$$K_r = \frac{\theta - \theta_r}{\theta_s - \theta_r} \tag{4.2}$$

where h_b and h_a are the parameters used to compute the water content and the relative hydraulic conductivity.
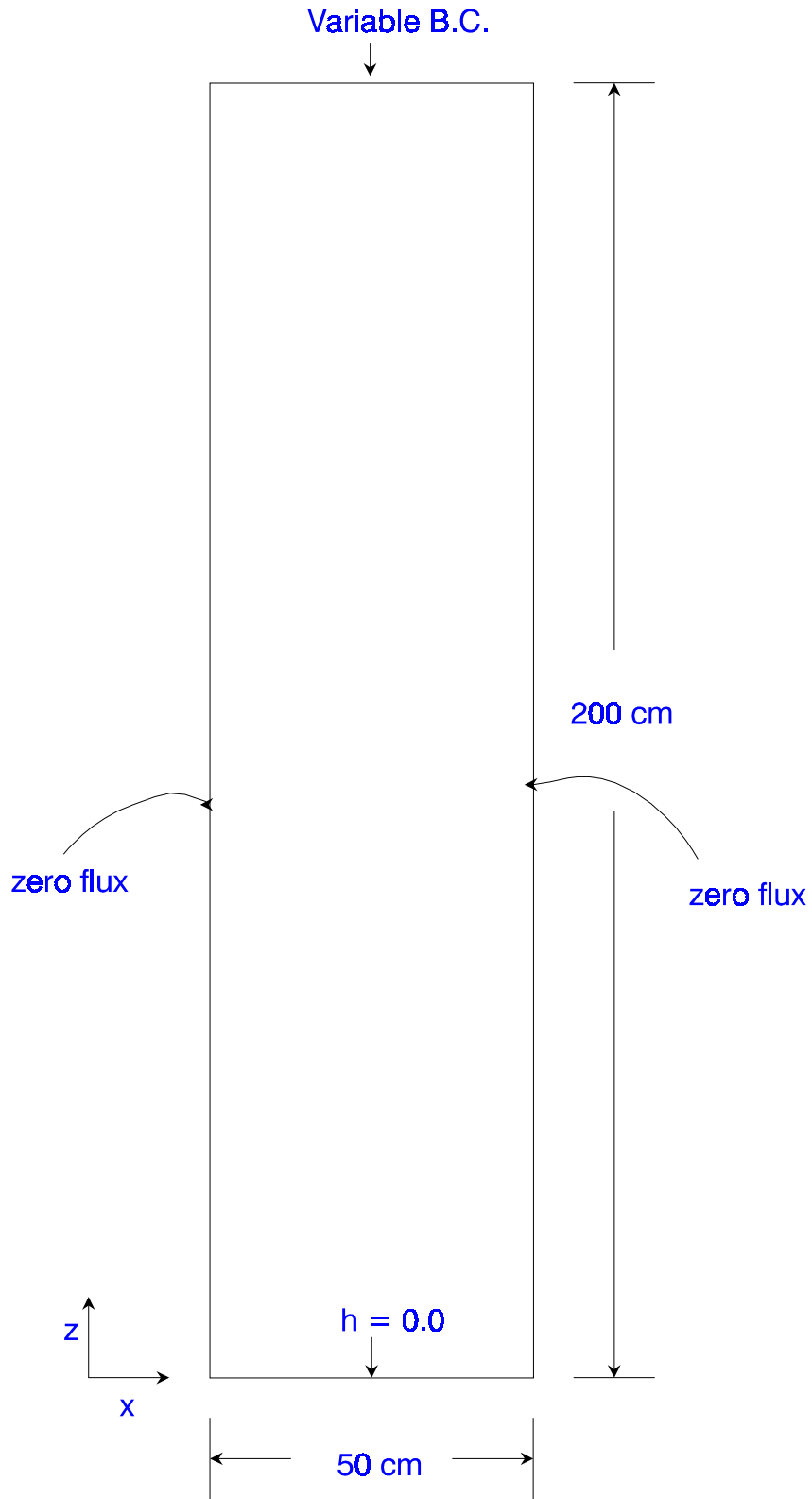
Figure 4.1 Problem definition for the one-dimensional transient flow in a soil column

The initial conditions are assumed: a pressure head of -90 cm is imposed on the top surface of the column, 0 cm on the bottom surface of the column, and -97 cm elsewhere. The boundary conditions are given as: no flux is imposed on the left, front, right, and back surfaces of the column; pressure head is held at 0 cm on the bottom surface; and variable condition is used on the top surface of the column with a ponding depth of zero, minimum pressure of -90 cm, and a rainfall of 5 cm/day for the first ten days and a potential evaporation of 5 cm/day for the second ten days.

The region of interest, i.e. the whole column, will be discretized with 1 x 40 = 40 elements with element size = 50 x 5 cm, resulting in 2 x 41 = 82 node points. A variable time-step size is used. The initial time step size is 0.05 days and each subsequent time-step size is increased by 0.2 times with a maximum time-step size not greater than 1.0 day. Because there is an abrupt change in the flux value from 5 cm/day (infiltration) to -5 cm/day (evaporation) imposed on the top surface at day 10, the time-step size is automatically reset to 0.05 day on tenth day. A 20 day simulation will be made with the revised FEMWATER. With the time-step size described above, 44 time-steps will be needed. The pressure head tolerance for nonlinear iteration is $2 \cdot 10^{-2}$ cm, and the relaxation factors is set equal to 0.5.

To execute the problem, the maximum control-integers in the MAIN program should be specified as:

```
PARAMETER(MAXELK=40,MAXNPK=82,MXBESK=82,MXBNPK= 82,MAXBWK=7)
PARAMETER(MXJBDK=9,MXKBDK=4,MXADNK=MAXNPK+1)
PARAMETER(MXMATK=1,MXSPPK=4,MXMPPK=10)
PARAMETER(MXNTIK=44,MXNDTK=3)

PARAMETER(MXSELF=1,MXSPRF=1,MXSDPF=1,MXWNPF=1,MXWPRF=1,MXWDPF=1)
PARAMETER(MXCNPF=1,MXCESF=1,MXCPRF=1,MXCDPF=1)
PARAMETER(MXNNPF=1,MXNESF=1,MXNPRF=1,MXNDPF=1)
PARAMETER(MXRESF=1,MXRNPF=2,MXRPRF=1,MXRDPF=4)
PARAMETER(MXDNPF=2,MXDPRF=1,MXDDPF=2)

PARAMETER(MXSELT=1,MXSPRT=1,MXSDPT=1,MXWNPT=1,MXWPRT=1,MXWDPT=1)
PARAMETER(MXCNPT=1,MXCEST=1,MXCPRT=1,MXCDPT=1)
PARAMETER(MXNNPT=1,MXNEST=1,MXNPRT=1,MXNDPT=1)
PARAMETER(MXVEST=1,MXVNPT=1,MXVPRT=1,MXVDPT=1)
PARAMETER(MXDNPT=1,MXDPRT=1,MXDDPT=1)
```

```
      PARAMETER(MXNCCK=1)

      PARAMETER(MXKGLDK=1,MXLSVK=1,MXMSVK=1,MXNDBK=1)
      PARAMETER(MXNEPK=1,MXEPWK=1)
      PARAMETER(MXNPWK=1,MXELWK=1,MXNPWS=1,MXELWS=1)
      PARAMETER(MXNPFGK=1,MXKGLK=1)
```

To reflect the soil property function given by Eqs. (4.1) and (4.2), we have to modify the

subroutine SPFUNC. A segment of the code in the subroutine SPFUNC must be modified as:

```
C %%%% EXAMPLE 1
C
      WCR=SPP(1,MTYP,1)
      WCS=SPP(2,MTYP,1)
      HAA=SPP(3,MTYP,1)
      HAB=SPP(4,MTYP,1)
CC
CC ------- SATURATED CONDITION
CC
C
C %%%% EXAMPLE 1
C
      IF(HNP.GT.0.0) GO TO 700
C
       TH=WCS
       IF(ISP.EQ.1)GOTO 900
       DTH=0.0D0
       AKHCX=SATKX*RHOM/AMU
       AKHCZ=SATKZ*RHOM/AMU
       AKHCXZ=SATKXZ*RHOM/AMU
       GOTO 900
CC
CC------- UNSATURATED CASE
CC
  700    CONTINUE
C
C %%%% EXAMPLE 1
C

      THMKG=WCs-(WCS-WCR)*(-hnp-haa)/(hab-haa)
      TH=THMKG
      IF(ISP.EQ.1) GOTO 900
      rk=(thmkg-wcr)/(wcs-wcr)
      DTH=-(wcs-wcr)/(hab-haa)
CC
      IF(RK.LT.CNSTKR)RK=CNSTKR
```

```
          AKHCX=SATKX*rk*RHOM/AMU
          AKHCZ=SATKZ*rk*RHOM/AMU
          AKHCXZ=SATKXZ*rk*RHOM/AMU
        ENDIF
C
  900  RETURN
      END
```

Input and Output for Problem No. 1

       With the above descriptions, the input data can be prepared according to the instructions given

in Appendix A. The input parameters are shown in Table 4.1 and the input data are given in Table 4.2.

The simulation results are identical to those obtained with FEMWATER. Thus, the flow module of

2DFATMIC is verifiied. To save space, output is available in electronic form.

Table 4.1 The list of input parameters for Problem 1

Parameters	Notation in the data input guide	Value	Unit	Data set
number of points	NNP	82	dimensionless	3. A.
Δx	XAD	50	cm	3. B.
Δz	ZAD	5	cm	3. B.
$K_{s,zz}$	PROPf(1,2)	10	cm/day	6. B.
θ_r	SPP(1,1,1)	0.15	dimensionless	7. B.
θ_s	SPP(2,1,1)	0.45	dimensionless	7. B.
h_a	SPP(3,1,1)	0	cm	7. B.
h_b	SPP(4,1,1)	-100	cm	7. B.
initial time-step size	DELT	0.05	day	8. B.
time-step size increment percentage	CHNG	0.2	dimensionless	8. B.
maximum time-step size	DELMAX	1	day	8. B.
no. of times to reset time-step size	NDTCHG	1	dimensionless	8. A.
time to reset time-step size	TDTCH(1)	10	day	8. E.
total simulation time	TMAX	22	day	8. B.
no. of time-steps	NTI	44	dimensionless	8. A.
tolerance for nonlinear iteration	TOLBh	2×10^{-2}	cm	9. B.
relaxation factor for nonlinear iteration	OMEh	0.5	dimensionless	9. B.
ρ_w	PROP(I,4)	1.0	g/cm^3	6.B.
μ_w	PROP(I,9)	9483.264	g/cm/day	6.B.
g	PROP(I,5)	7.316×10^{12}	cm/day^2	6.B.

Table 4.2 Input Data Set for Problem 1

```
     1     Test of 2DFATMIC, flow only, FEMWATER.EX1, cm,g,day
     1  1  1  1  10 100  0.5  1  iitr,ichng,inter,ibuf,imod,nitrht,omeht,igeom
C ***** data set 2
0 0  0   0  0  1 0  0 12 0 IMID,IWET,ILUMP,IOPTIM,IPNTS,IVML,NSTRh,nstrt,iquar,idis
C ***** data set 3
82                        NNP
1  40  2   0.0  0.0  0.0     X
2  40  2   5.0  0.0  0.0     COORDINATE
0   0  0   0.0  0.0  0.0
1  40  2   0.0  5.0  0.0     Z
2  40  2   0.0  5.0  0.0     COORDINATE
0   0  0   0.0  0.0  0.0
C ***** data set 4
40  1  0                  NEL  NMAT  KCP
1  1  2  4  3  1  1  40        IE
C ****** data set 5
0                         NCM
C ***** data set 6
10  1  1                  NMPPM NCC  IRXN
0.00 10.0 0.0  1.0d0 7.316D12  0.36   0.36   3.6D-5  948.0d0 1.2    PROP
0.5                          DKD
0.5                          TRANC
1.0                          DINTS
0.0                          RHOMU
C ***** data set 7
0  4                              KSP  NSPPM
0.15  0.45  0.0  -1.0d2           THKCAH
0.0   0.0   0.0   0.0  0.0        AKPROP
C ***** data set 8
44 3                              NTI,NDTCHG
0.05  0.2  1.0D0  20.0d0           DELT,CHNG,DELMAX,TMAX
555050500050005005000005550505050005000500500005  KPR
111010100010001001000011101010001000100100001  KDSK
1.0d1  2.0d1  1.0D38               TDTCH
C ***** data set 9
20  50  300  0  1  1                 NCYL,NITER,NPITER,KSTR,KSS,KGRAV
2.0D-2  2.0D-2  1.0  0.5  0.5 0.0      TOLAH,TOLBH,WF,OMEH,OMIH,cnstkr
C ***** data set 10
1  1   1   0.0  0.0   0.0            I.C.
3  77  1  -97.0  0.0  0.0
81  1  1  -90.0  0.0  0.0
0  0  0    0.0  0.0  0.0
C ***** data set 11
0  0  0 0                         NSELH,NSPRH,NSDPH,KSAIH
C ***** data set 12
0  0  0 0                         NWNPH,NWPRH,NWDPH,KWAIH
C ***** data set 13
1   2   1  4 0                    NRES,NRNP,NRPR,NRDP,KRAIH
0.0  5.0  10.0  5.0  10.001  -5.0  1.0D38  -5.0   TRF,RF OF PROFILE 1
1  1  1   81   1                  IRTYP(1,2)
0   0  0   0    0
1  1 1    1    0                  IRTYP(1,1)
0   0  0    0    0
1  1   1  0.0D0  0.0D0   0.0
0  0   0  0.0D0  0.0D0   0.0      RSVAB(1,1) --- HCON
1  1   1  -90.0D0  0.0D0  0.0
0  0   0  0.0D0  0.0D0   0.0      RSVAB(1,2) --- HMIN
1  0  40  1  2   0  1  1       MI,NSEQ,M,IS1,IS2,MIAD,MAD,IS1AD,IS2AD
0  0  0  0  0  0  0  0  0
C ***** data set 14
2  1   2   0                     NDNPH,NDPRH,NDDPH,KDAIH
0.0  0.0  1.0D38  0.0             THDBF,HDBF
1   1 1 1   1                     IDTYP(1,2)
0   0  0 0   0
1   1 1 1   0                     IDTYP(1,1)
0   0  0 0   0
```

```
C ****** data set 15
0  0  0  0  0                        NCES,NCNP,NCPR,NCDP,KCAIH
C ****** data set 16
0  0  0  0  0                        NNES,NNNP,NNPR,NNDP,KNAIH
0
```

4.2 Problem No. 2: One-Dimensional Column Transport

<u>Statement of Problem No. 2</u>

A simple problem is presented here to illustrate the application of this document.   This is a one-dimensional solute transport and biotransformation between x = 0 and x = 50.0 cm (Figure 4.2). Initially, the concentrations of microbe #1 (C_1) and microbe #2 (C_2) are zero, and those of microbe #3 (C_3), substrate (C_s), oxygen (C_o), nitrate (C_n) and nutrient (C_p) are 3.64×10^{-3}, 1.0×10^{-3}, 2.0×10^{-3}, 5.0×10^{-3} and 1.0×10^2 mg/cm^3 throughout the region of interest. This case is to demonstrate that multiple electron acceptor respiration can occur and the simulation is done with excess nutrient. The concentration at x = 0.0 is maintained at $C_1 = C_2 = 0$, $C_3 = 3.64 \times 10^{-3}$, $C_s = 2.0 \times 10^{-2}$, $C_o = 2.0 \times 10^{-3}$, $C_n = 5.0 \times 10^{-3}$, and $C_p = 1.0 \times 10^2$ mg/cm^3. The natural condition of zero gradient flux is imposed at x = 50.0.

A bulk density of 1.2, a dispersivity of 0.5, an effective porosity of 0.4 (not used in the program) are assumed. A specific discharge (Darcy velocity) of 10.0 is assumed and a moisture content of 1.0 is used. The retardation factors of microbe #1, microbe #2, microbe #3, substrate, oxygen, nitrate, and nutrient are 10^5, 10^5, 10^5, 1.1, 1.0, 1.0 and 2.2, respectively.

For numerical simulation the region is divided into 100 elements of equal size of 0.5. A time- step size of 0.1 is used and 80 time-step simulation is made. For this discretization, mesh Peclet number is $P_e = 1$ and Courant number $C_r = 2$.
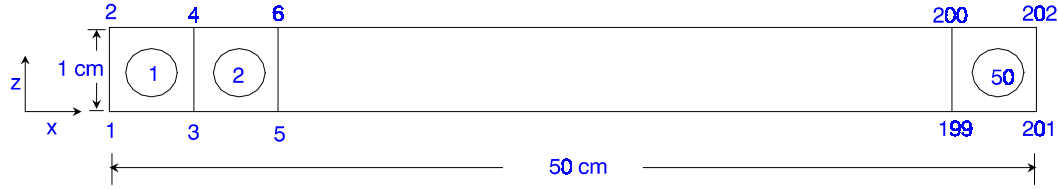
Figure 4.2 Region of Interest for Example 2

To execute the problem, the maximum control-integers in the MAIN must be specified as

```
PARAMETER(MAXELK=100,MAXNPK=202,MXBESK=202,MXBNPK= 202,MAXBWK=11)
PARAMETER(MXJBDK=11,MXKBDK=4,MXADNK=MAXNPK+3000)
PARAMETER(MXMATK=1,MXSPPK=4,MXMPPK=10)
PARAMETER(MXNTIK= 80,MXNDTK=3)

PARAMETER(MXSELF=1,MXSPRF=1,MXSDPF=1,MXWNPF=1,MXWPRF=1,MXWDPF=1)
PARAMETER(MXCNPF=1,MXCESF=1,MXCPRF=1,MXCDPF=1)
PARAMETER(MXNNPF=1,MXNESF=1,MXNPRF=1,MXNDPF=1)
PARAMETER(MXRESF=1,MXRNPF=1,MXRPRF=1,MXRDPF=1)
PARAMETER(MXDNPF=1,MXDPRF=1,MXDDPF=1)

PARAMETER(MXSELT=1,MXSPRT=1,MXSDPT=1,MXWNPT=1,MXWPRT=1,MXWDPT=1)
PARAMETER(MXCNPT=1,MXCEST=1,MXCPRT=1,MXCDPT=1)
PARAMETER(MXNNPT=1,MXNEST=1,MXNPRT=1,MXNDPT=1)
PARAMETER(MXVEST=1,MXVNPT=2,MXVPRT=1,MXVDPT=2)
PARAMETER(MXDNPT=2,MXDPRT=6,MXDDPT=2)

PARAMETER(MXNCCK=7)

PARAMETER(MXKGLDK=3000,MXLSVK=1,MXMSVK=1,MXNDBK=1200)
PARAMETER(MXNEPK=1,MXEPWK=1)
PARAMETER(MXNPWK=36,MXELWK=25,MXNPWS=36,MXELWS=25)
PARAMETER(MXNPFGK=35000,MXKGLK=14000)
```

Input and Ouput for Problem No. 2

Table 4.3 lists the input parameters and Table 4.4 shows the input data set for this example. To save space, the output isavailable in electronic form.

85

Table 4.3 The list of input parameters for Problem 2

Parameters	Notation in the data input guide	Value	Unit	Data set
number of points	NNP	202	dimensionless	3. A.
$\triangle x$	XAD	0.5	cm	3. B.
$\triangle z$	ZAD	1	cm	3. B.
$K_{d.1}$	RKD(1)	10^5	cm^3/mg	6. C.
$K_{d.2}$	RKD(2)	10^5	cm^3/mg	6. C.
$K_{d.3}$	RKD(3)	10^5	cm^3/mg	6. C.
$K_{d.s}$	RKD(4)	0.1	cm^3/mg	6. C.
$K_{d.o}$	RKD(5)	0	cm^3/mg	6. C.
$K_{d.n}$	RKD(6)	0	cm^3/mg	6. C.
$K_{d.p}$	RKD(7)	1.2	cm^3/mg	6. C.
ρ_b	PROPt(1,10)	1.0	mg/cm^3	6. B.
α_L	PROPt(1,6)	0.5	cm	6. B.
$\mu_o^{(1)}$	GRATE(1)	0.0	1/day	19. A.
$\mu_n^{(2)}$	GRATE(2)	0.0	1/day	19. A.
$\mu_o^{(3)}$	GRATE(3)	0.62	1/day	19. A.
$\mu_n^{(3)}$	GRATE(4)	0.58	1/day	19. A.
$Y_o^{(1)}$	YCOEFF(1)	0.45	mg/mg	19. B.
$Y_n^{(2)}$	YCOEFF(2)	0.5	mg/mg	19. B.
$Y_o^{(3)}$	YCOEFF(3)	0.45	mg/mg	19. B.
$Y_n^{(3)}$	YCOEFF(4)	0.5	mg/mg	19. B.
$K_{so}^{(1)}$	RTARDS(1)	0.04	mg/cm^3	19. C.
$K_{sn}^{(2)}$	RTARDS(2)	0.04	mg/cm^3	19. C.
$K_{so}^{(3)}$	RTARDS(3)	0.04	mg/cm^3	19. C.
$K_{sn}^{(3)}$	RTARDS(4)	0.04	mg/cm^3	19. C.
$K_o^{(1)}$	RTARDO(1)	7.7×10^{-4}	mg/cm^3	19. D.
$K_n^{(2)}$	RTARDO(2)	2.6×10^{-3}	mg/cm^3	19. D.

$K_o^{(3)}$	RTARDO(3)	7.7×10^{-4}	mg/cm^3	19. D.
$K_n^{(3)}$	RTARDO(4)	2.6×10^{-3}	mg/cm^3	19. D.
$K_{po}^{(1)}$	RTARDN(1)	1.0×10^{-3}	mg/cm^3	19. E.
$K_{pn}^{(2)}$	RTARDN(2)	1.0×10^{-3}	mg/cm^3	19. E.
$K_{po}^{(3)}$	RTARDN(3)	1.0×10^{-3}	mg/cm^3	19. E.
$K_{pn}^{(3)}$	RTARDN(4)	1.0×10^{-3}	mg/cm^3	19. E.
$\gamma_o^{(1)}$	SCOEFF(1)	1.4	dimensionless	19. F.
$\gamma_n^{(2)}$	SCOEFF(2)	2.2	dimensionless	19. F.
$\gamma_o^{(3)}$	SCOEFF(3)	1.4	dimensionless	19. F.
$\gamma_n^{(3)}$	SCOEFF(4)	2.2	dimensionless	19. F.
$\alpha_o^{(1)}$	ECOEFF(1)	0.0402	dimensionless	19. G.
$\alpha_n^{(2)}$	ECOEFF(2)	0.1	dimensionless	19. G.
$\alpha_o^{(3)}$	ECOEFF(3)	0.0402	dimensionless	19. G.
$\alpha_n^{(3)}$	ECOEFF(4)	0.1	dimensionless	19. G.
$\lambda_o^{(1)}$	DCOEFF(1)	0.02	1/day	19. H.
$\lambda_n^{(2)}$	DCOEFF(2)	0.02	1/day	19. H.
$\lambda_o^{(3)}$	DCOEFF(3)	0.004	1/day	19. H.
$\lambda_n^{(3)}$	DCOEFF(4)	0.004	1/day	19. H.
$\Gamma_o^{(1)}$	SATURC(1)	7.7×10^{-4}	mg/cm^3	19. I.
$\Gamma_n^{(2)}$	SATURC(2)	2.6×10^{-3}	mg/cm^3	19. I.
$\Gamma_o^{(3)}$	SATURC(3)	7.7×10^{-4}	mg/cm^3	19. I.
$\Gamma_n^{(3)}$	SATURC(4)	2.6×10^{-3}	mg/cm^3	19. I.
$\epsilon_o^{(1)}$	PCOEFF(1)	0.122	dimensionless	19. J.
$\epsilon_n^{(2)}$	PCOEFF(2)	0.122	dimensionless	19. J.
$\epsilon_o^{(3)}$	PCOEFF(3)	0.122	dimensionless	19. J.
$\epsilon_n^{(3)}$	PCOEFF(4)	0.122	dimensionless	19. J.
K_c	COFK	1.1×10^{-4}	mg/cm^3	19. K.
maximum time-step size	DELMAX	0.1	day	8. B.
no. of times to reset time-step size	NDTCHG	0	dimensionless	8. A.

total simulation time	TMAX	8	day	8. B.
no. of time-steps	NTI	80	dimensionless	8. A.
tolerance for nonlinear iteration	TOLBt	1×10^{-4}	dimensionless	17. B.
relaxation factor for nonlinear iteration	OMEt	1.0	dimensionless	17. B.

Table 4.4 Input Data Set for Problem 2

```
   1 adv-dis, Multi-electron acceptor, excess nutrient, VB, cm, mg, day
   1  1  1  1  1 100  0.5 1   iitr,ichng,inter,ibuf,imod,nitrht,omeht,igeom
C ***** DATA SET 2 : OPTION PARAMETERS FOR BOTH FLOW AND TRANSPORT
0 0  1  1  0   0 0 0 12 0 IMID,IWET,ILUMP,IOPTIM,IPNTS,IVML,NSTRH,NSTRT,iquar,idis
C ***** DATA SET 3 : NODAL COORDINATE
202             NNP
1 100  2   0.0D0   0.5D0  0.0D0      X
2 100  2   0.0D0   0.5D0  0.0D0      X
0  0  0   0.0D0   0.0D0  0.0D0
1 100  2   0.0D0   0.0D0  0.0D0      Z
2 100  2   1.0D0   0.0D0  0.0D0      COORDINATE
0  0  0  0.0  0.0   0.0
C ***** DATA SET 4 : ELEMENT INCIDENCES
100 1  0                NEL   NMAT  KCP
1  1  3   4   2  1  1  100      IE
C ***** DATA SET 5 : MATERIAL TYPE CORRECTION
0                  NCM
C ***** DATA SET 6 : MATERIAL PROPERTIES
10  7  1               NMPPM  NCC  IRXN
0.00 10.0 0.0  1.0d3 7.316D12  0.5  0.0  0.0  9.48d-5  1.0  PROP
1.0d5  1.0d5  1.0d5   0.1  0.0  0.0  1.2     DKD
0.0  0.0   0.0  0.0  0.0  0.0  0.0     TRANC
 1.0  1.0  1.0  1.0  1.0  1.0  1.0     DINTS
 0.0  0.0  0.0  0.0  0.0  0.0  0.0     RHOMU
C ****** DATA SET 7 : SOIL PROPERTIES
0  4                       KSP  NSPPM
0.15  0.45  0.0  -1.0d2        SPP(J,I,1)
0.0   0.0   0.0   0.0  0.0     SPP(J,I,2)
C ***** DATA SET 8 : TIME CONTROL PARAMETERS
80 1                        NTI,NDTCHG
0.1D0  0.0  9.0D1   90.0D0        DELT,CHNG,DELMAX,TMAX
555550 0 0 000 00 00          50000
5
1 0000 0 0 000 00 00          10000
1
            1.0D38                TDTCH
C ***** DATA SET 17 : BASIC REAL AND INTEGER PARAMETERS FOR TRANSPROT
50  999 1  1   -1  1       nitert,npitert,kstrt,ksst,kvit,miconf
1.0d-3  1.0d-4  1.0  0.0  1.0d0  1.0d0  1.0    tolat,tolbt,wt,wv,omet,omit,aphag
1  1  0  5  1  5  1   1  1  2 IZOOM,IDZOOM,IEPC,NXA,NZA,NXD,NZD,NXW,NZW,IDETQ
0.0005  0.0005             ADPEPS, ADPARM
C ***** DATA SET 18 : INITIAL/PRE-INITIAL CONDITIONS FOR TRANSPORT
1  201 1  0.0D0   0.0  0.0          IC FOR Microbe #1
0   0  0  0.0  0.0  0.0
1  201  1  0.0d0    0.0  0.0          IC FOR microbe #2
0   0  0  0.0  0.0  0.0
1  201  1 3.64d-8 0.0  0.0          IC FOR Microbe #3
0   0  0  0.0  0.0  0.0
```

```
1  201  1  1.0D-3  0.0  0.0              IC FOR Substrate
0   0   0  0.0  0.0  0.0
1  201  1  2.0D-3  0.0  0.0              IC FOR Oxygen
0   0   0  0.0  0.0  0.0
1  201  1  5.0d-3  0.0  0.0              IC FOR Nitrate
0   0   0  0.0  0.0  0.0
1  201  1  1.0D2  0.0  0.0               IC FOR Nutrient
0   0   0  0.0  0.0  0.0
C ***** DATA SET 19 : MICROBE-CHEMICAL INTERACTION CONSTANTS
 0.0  0.0  0.62 0.58                     GRATE
 0.45  0.5 0.45 0.5                      YCOEFF
4.0D-2  4.0D-2  4.0D-2  4.0D-2           RTARDS   Kso, Ksn
7.7D-4  2.6D-3  7.7D-4  2.6D-3           RTARDO   Ko, Kn
1.0d-3  1.0D-3  1.0D-3  1.0D-3           RTARDN   Kpo, Kpn
1.4  2.2    1.4    2.2                   SCOEFF   gammao, gamman
4.02D-2 1.0D-1 4.02D-2 1.0D-1            ECOEFF   alphao, alphan
0.02 0.02 0.004    0.004                 DCOEFF   lambdao, lambdan
7.7D-4  2.6D-3    7.7D-4 2.6D-3          SATURC   GAMMAo, GAMMAn
1.22D-1 1.22D-1 1.22D-1 1.22D-1          PCOEFF   Epsilon
  1.1d-4                                 COFK
C ***** DATA SET 20 : ELEMENT SOURCE/SINK
0  0  0  0                               NSEL,NSPR,NSDP,ksai
C ***** DATA SET 21 : WELL SOURCE/SINK
0  0  0  0                               NWNP,NWPR,NSDP,kwai
C ***** DATA SET 22 : VARIABLE B.C.
1  2  1  2  0                            NVES,NVNP,NVPR,NVDP,kdai
0.0  0.0    1.0d38  0.0d0
1  0  1  1  0                            IvTYP FOR Microbe #1
0  0  0  0  0
1  0  1  1  0                            IVTYP for Microbe #2
0  0  0  0  0
1  0  1  1  0                            IVTYP for Microbe #3
0  0  0  0  0
1  0  1  1  0                            IVTYP for Substrate
0  0  0  0  0
1  0  1  1  0                            IVTYP for Oxygen
0  0  0  0  0
1  0  1  1  0                            IVTYP for Nitrate
0  0  0  0  0
1  0  1  1  0                            IVTYP for Nutrient
0  0  0  0  0
1  1  1  201 1                           NPVB
0  0  0  0   0
1  0  100 1 2  1 1   1  1                ISV
0  0  0   0 0  0  0  0  0
C ***** DATA SET 23 : DIRICHLET B.C.
2   6  2  0                              DIRICHLET BOUNDARY CONDITION
0.0  2.0D-2 1.0D38  2.0D-2    TCDBF  CDBF    Profile 1
0.0  2.0D-3 1.0D38  2.0D-3                   Profile 2
0.0  3.64d-8 1.0D38  3.64d-8                 Profile 3
0.0  1.0D2 1.0D38  1.0D2                     Profile 4
0.0  5.0d-3  1.0d38  5.0d-3                  Profile 5
0.0  0.0    1.0d38  0.0d0                    Profile 6
1  1  1  6  0                            IDTYP for Microbe #1
0  0  0  0  0
1  1  1  6  0                            IDTYP for Microbe #2
0  0  0  0  0
1  1  1  3  0                            IDTYP for Microbe #3
0  0  0  0  0
1  1  1  1  0                            IDTYP for substrate
0  0  0  0  0
1  1  1  2  0                            IDTYP for oxygen
0  0  0  0  0
1  1  1  5  0                            IDTYP for nitrate
0  0  0  0  0
1  1  1  4  0                            IDTYP for nutrient
0  0  0  0  0
1  1  1  1  1                                    NPDB
0  0  0  0  0
```

```
C ***** DATA SET 24 : CAUCHY B.C.
0   0  0  0  0                           nces,ncnp,ncpr,ncdp,kcai
C ***** DATA SET 25 : NEUMANN B.C.
0   0  0  0  0                           nnes,nnnp,nnpr,nndp,knai
C ***** DATA SET 26 : HYDROLOGICAL VARIABLES
1 201 1 10.0  0.0  0.0             VELOC(NNP,1)
0   0  0  0.0   0.0   0.0
1 201 1   0.0   0.0   0.0          VELOC(NNP,2)
0   0  0  0.0   0.0   0.0
1 99   1  1.0d0  0.0                TH(1..4,NEL)
0   0  0  0.0   0.0
0
```

<u>Simulation Results</u>

The results at t = 4 days and t = 8 days are shown as follows. These results are similar to those obtained

in Widdowson, et. al. (1988). Thus, the transport module of 2DFATMIC is indirectly verified.
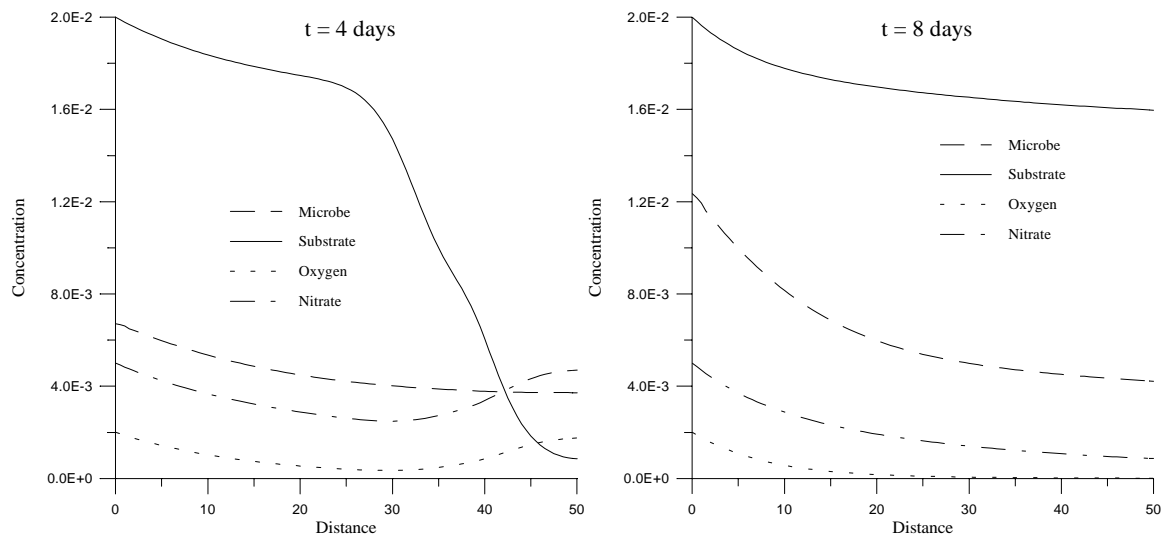
Figure 4.3 The concentration profiles of biomass, substrate, oxygen, and nitrate at
t = 4 days and t = 8 days

4.3 Problem No. 3: Two-Dimensional Coupled Flow and Multicomponent Transport Problem

Statement of Problem No. 3

This problem is presented in "Denitrification in nonhomogeneous laboratory scale aquifers: 5: user's manual for the mathematical model LT3VSI" by G.A. Bachelor et al. reported in 1990. The example aquifer used for this problem is 1.4 meters long in the X direction, 1.6 meters thick in the Z direction, and shown in Figure 4.4. This aquifer has 8 different materials, one injection well at (0.1,0.1) and one extraction well at (1.3,1.5). The hydrological and microbial dynamical data are all from this report. The initial condition of the flow field is obtained by simulating steady state of flow field without sources and sinks. Then the flow field and concentration distribution are updated at each time step.

There are two types of microbes included in the system, say microbe #1 and microbe #3 with 1.77×10^{-4} Kg/m^3 initially. The initial concentrations of the chemicals are 5×10^{-3} Kg/m^3 of substrate, 5×10^{-3} Kg/m^3 of oxygen, 5×10^{-3} Kg/m^3 of nitrate, and 3×10^{-3} Kg/m^3 of nutrient. The daily injection and withdrawal rates of water are 3.75×10^{-3} and 3.75×10^3 m^3, respectively. The total hydraulic head is 1.0 m at the upstream boundary AB (Figure 4.4) and 0.0 m at the downstream boundary CD (Figure 4.4). For transport simulation, variable boundary condition is implemented at the downstream boundary CD (Figure 4.4) and 1.77×10^{-4} Kg/m^3 of microbe #1, 1.77×10^{-4} Kg/m^3 of microbe #3, 1.5×10^{-2} Kg/m^3 of substrate, 5.0×10^{-3} Kg/m^3 of oxygen, 5.0×10^{-3} Kg/m^3 of nitrate, and 3.0×10^{-3} Kg/m^3 of nutrient influents through the upstream boundary. Because microbe #2 does not exist in this environment, the initial and boundary conditions for this component are set to zero in this simulation. This problem is set up for 4 day simulation.
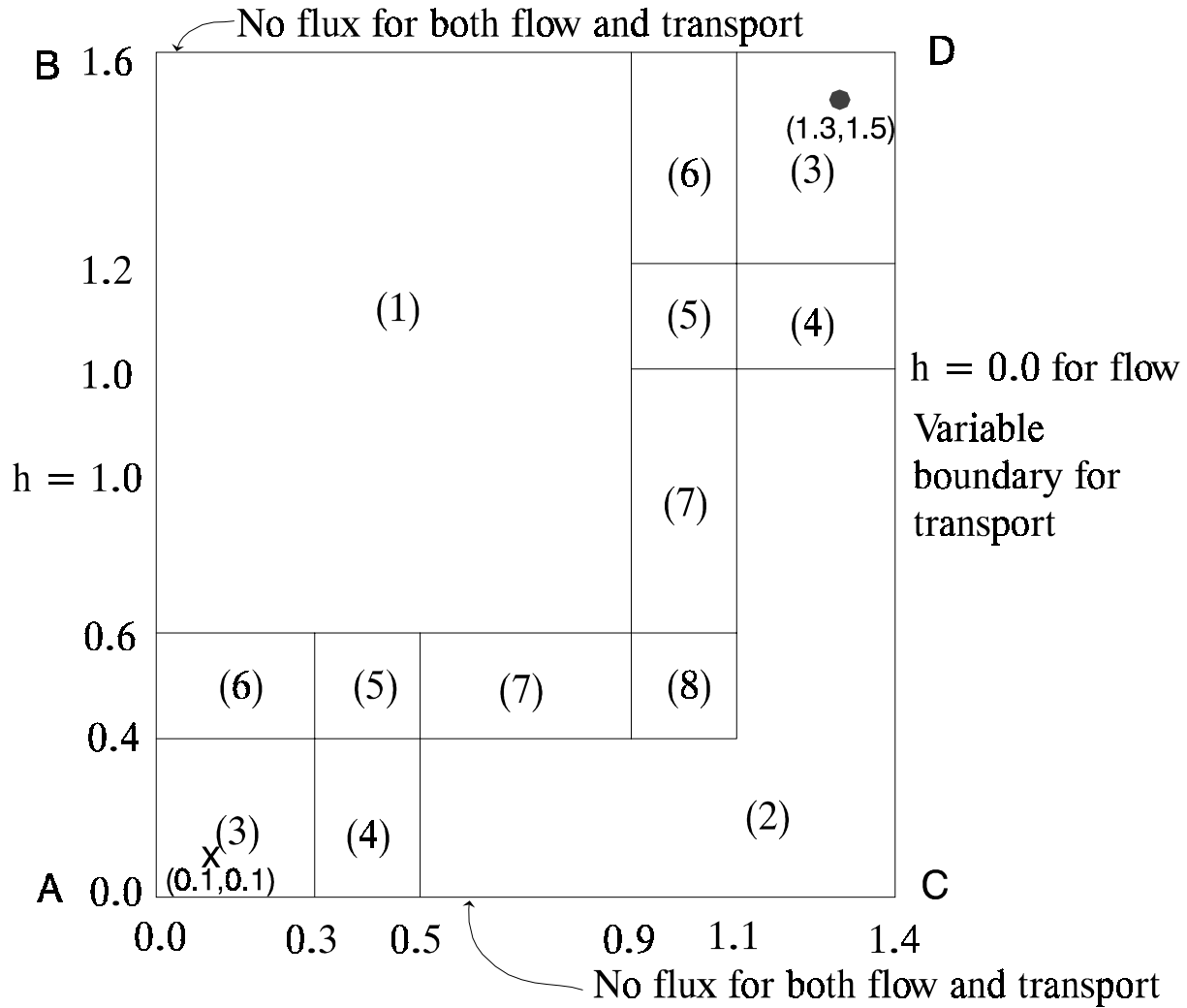
Figure 4.4 The region of Problem 3 with 8 different material properties

For numerical simulation the region is divided into 14 x 16 = 224 square elements of size = 0.1 by 0.1, and 15 x 17 = 255 nodes. A time-step size of 0.05 is used and 80 time-step simulation is made.

To execute the problem, the maximum control-integers in the MAIN must be specified as

```
PARAMETER(MAXELK=224,MAXNPK=255,MXBESK=60,MXBNPK=60,MAXBWK=33)
PARAMETER(MXJBDK=17,MXKBDK=4,MXADNK=MAXNPK+4200)
PARAMETER(MXMATK=8,MXSPPK=2,MXMPPK=10)
PARAMETER(MXNTIK=80,MXNDTK=3)
```

PARAMETER(MXSELF=1,MXSPRF=1,MXSDPF=1,MXWNPF=2,MXWPRF=2,MXWDPF=3)
PARAMETER(MXCNPF=1,MXCESF=1,MXCPRF=1,MXCDPF=1)
PARAMETER(MXNNPF=1,MXNESF=1,MXNPRF=1,MXNDPF=1)
PARAMETER(MXRESF=1,MXRNPF=1,MXRPRF=1,MXRDPF=1)
PARAMETER(MXDNPF=34,MXDPRF=2,MXDDPF=2)

PARAMETER(MXSELT=1,MXSPRT=1,MXSDPT=1,MXWNPT=2,MXWPRT=2,MXWDPT=3)
PARAMETER(MXCNPT=1,MXCEST=1,MXCPRT=1,MXCDPT=1)
PARAMETER(MXNNPT=1,MXNEST=1,MXNPRT=1,MXNDPT=1)
PARAMETER(MXVEST=16,MXVNPT=17,MXVPRT=1,MXVDPT=2)
PARAMETER(MXDNPT=17,MXDPRT=5,MXDDPT=2)

PARAMETER(MXNCCK=7)

PARAMETER(MXKGLDK=4200,MXLSVK=700,MXMSVK=150,MXNDBK=500)
PARAMETER(MXNEPK=1,MXEPWK=1)
PARAMETER(MXNPWK=36,MXELWK=25,MXNPWS=36,MXELWS=25)
PARAMETER(MXNPFGK=7100,MXKGLK=17500)

Input and Ouput for Problem No. 3

Table 4.5 shows the input parameters used for this example. Table 4.6 shows the input data set

for the transient simulation of the coupled problem. The output results are given isavailable in electronic

form.

Table 4.5 The list of input parameters for Problem 3

Parameters	Notation in the data input guide	Value	Unit	Data set
number of points	NNP	510	dimensionless	3. A.
Δx	XAD	0.1	m	3. B.
Δz	ZAD	0.1	m	3. B.
no. of materials	NMAT	8	dimensionless	4. A.
$\mu_o^{(1)}$	GRATE(1)	4.0	1/day	19. A.
$\mu_n^{(2)}$	GRATE(2)	0.0	1/day	19. A.
$\mu_o^{(3)}$	GRATE(3)	4.0	1/day	19. A.
$\mu_n^{(3)}$	GRATE(4)	2.5	1/day	19. A.

$Y_o^{(1)}$	YCOEFF(1)	0.4	Kg/Kg	19. B.
$Y_n^{(2)}$	YCOEFF(2)	0.17	Kg/Kg	19. B.
$Y_o^{(3)}$	YCOEFF(3)	0.4	Kg/Kg	19. B.
$Y_n^{(3)}$	YCOEFF(4)	0.17	Kg/Kg	19. B.
$K_{so}^{(1)}$	RTARDS(1)	0.018	Kg/m^3	19. C.
$K_{sn}^{(2)}$	RTARDS(2)	0.018	Kg/m^3	19. C.
$K_{so}^{(3)}$	RTARDS(3)	0.018	Kg/m^3	19. C.
$K_{sn}^{(3)}$	RTARDS(4)	0.018	Kg/m^3	19. C.
$K_o^{(1)}$	RTARDO(1)	3.0×10^{-5}	Kg/m^3	19. D.
$K_n^{(2)}$	RTARDO(2)	2.0×10^{-5}	Kg/m^3	19. D.
$K_o^{(3)}$	RTARDO(3)	3.0×10^{-5}	Kg/m^3	19. D.
$K_n^{(3)}$	RTARDO(4)	2.0×10^{-5}	Kg/m^3	19. D.
$K_{po}^{(1)}$	RTARDN(1)	3.0×10^{-4}	Kg/m^3	19. E.
$K_{pn}^{(2)}$	RTARDN(2)	3.0×10^{-4}	Kg/m^3	19. E.
$K_{po}^{(3)}$	RTARDN(3)	3.0×10^{-4}	Kg/m^3	19. E.
$K_{pn}^{(3)}$	RTARDN(4)	3.0×10^{-4}	Kg/m^3	19. E.
$\gamma_o^{(1)}$	SCOEFF(1)	1.0	dimensionless	19. F.
$\gamma_n^{(2)}$	SCOEFF(2)	0.375	dimensionless	19. F.
$\gamma_o^{(3)}$	SCOEFF(3)	1.0	dimensionless	19. F.
$\gamma_n^{(3)}$	SCOEFF(4)	0.375	dimensionless	19. F.
$\alpha_o^{(1)}$	ECOEFF(1)	0.004	dimensionless	19. G.
$\alpha_n^{(2)}$	ECOEFF(2)	0.002	dimensionless	19. G.
$\alpha_o^{(3)}$	ECOEFF(3)	0.004	dimensionless	19. G.
$\alpha_n^{(3)}$	ECOEFF(4)	0.002	dimensionless	19. G.
$\lambda_o^{(1)}$	DCOEFF(1)	0.02	1/day	19. H.
$\lambda_n^{(2)}$	DCOEFF(2)	0.02	1/day	19. H.
$\lambda_o^{(3)}$	DCOEFF(3)	0.02	1/day	19. H.
$\lambda_n^{(3)}$	DCOEFF(4)	0.02	1/day	19. H.
$\Gamma_o^{(1)}$	SATURC(1)	3.0×10^{-5}	Kg/m^3	19. I.

$\Gamma_n^{(2)}$	SATURC(2)	2.0×10^{-5}	Kg/m^3	19. I.
$\Gamma_o^{(3)}$	SATURC(3)	3.0×10^{-5}	Kg/m^3	19. I.
$\Gamma_n^{(3)}$	SATURC(4)	2.0×10^{-5}	Kg/m^3	19. I.
$\epsilon_o^{(1)}$	PCOEFF(1)	0.05	dimensionless	19. J.
$\epsilon_n^{(2)}$	PCOEFF(2)	0.021	dimensionless	19. J.
$\epsilon_o^{(3)}$	PCOEFF(3)	0.05	dimensionless	19. J.
$\epsilon_n^{(3)}$	PCOEFF(4)	0.021	dimensionless	19. J.
K_c	COFK	1.1×10^{-4}	Kg/m^3	19. K.
no. of elements	NEL	224	dimensionless	4. A.
no. of materials to be corrected	NCM	134	dimensionless	5. A.
steady-state for flow	KSSh	0	dimensionless	9. A.
transient-state for transport	KSSt	1	dimensionless	17. A.
initial time-step size	DELT	0.05	day	8. B.
time-step size increment percentage	CHNG	0	dimensionless	8. B.
maximum time-step size	DELMAX	0.05	day	8. B.
no. of times to reset time-step size	NDTCHG	0	dimensionless	8. A.
total simulation time	TMAX	4	day	8. B.
no. of time-steps	NTI	80	dimensionless	8. A.
tolerance for flow nonlinear iteration	TOLAh	1×10^{-2}	m	9. B.
relaxation factor for flow nonlinear iteration	OMEh	1.0	dimensionless	9. B
tolerance for transport nonlinear iteration	TOLBt	1×10^{-4}	dimensionless	17. B.
relaxation factor for transport nonlinear iteration	OMEt	1.0	dimensionless	17. B.
ρ_w	PROP(I,4)	1000.0	Kg/m^3	6. B.
μ_w	PROP(I,9)	948.3264	Kg/m/day	6. B.

g	PROP(I,5)	7.316×10^{10}	m/day^2	6.B.

Table 4.6 Input Data Set for Problem 3

```
    1 advection-dispersion, flow & transport,multi-materials, Kg, m, day
    1  1  1   1 11   100 0.5 0 iitr,ichng,inter,ibuf,imod,nitrht,omeht,igeom
C ***** DATA SET 8: CONTROL PARAMETER
0 0  1  0  0   1 0   0 12 1 IMID,IWET,ILUMP,IOPTIM,IPNTS,IVML,NSTRH,NSTRT,iquar,idis
C ***** DATA SET 2 : NODAL COORDINATE
255                         NNP
1  16  15   0.0  0.0  0.0         X
2  16  15   0.1  0.0  0.0         X
3  16  15   0.2  0.0  0.0         X
4  16  15   0.3  0.0  0.0         X
5  16  15   0.4  0.0  0.0         X
6  16  15   0.5  0.0  0.0         X
7   16  15   0.6  0.0  0.0          X
8   16  15   0.7  0.0  0.0          X
9    16  15   0.8  0.0  0.0           X
10  16  15   0.9  0.0  0.0         X
11  16  15   1.0  0.0  0.0         X
12  16  15   1.1  0.0  0.0         X
13  16  15   1.2  0.0  0.0         X
14  16  15   1.3  0.0  0.0         X
15  16  15   1.4  0.0  0.0         X
0   0   0   0.0  0.0  0.0
1  16  15   0.0  0.1  0.0         Z
2  16  15   0.0  0.1  0.0         Z
3  16  15   0.0  0.1  0.0         Z
4  16  15   0.0  0.1  0.0         Z
5  16  15   0.0  0.1  0.0         Z
6  16  15   0.0  0.1  0.0         Z
7   16  15   0.0  0.1  0.0          Z
8   16  15   0.0  0.1  0.0          Z
9    16  15   0.0  0.1  0.0           Z
10  16  15   0.0  0.1  0.0         Z
11  16  15   0.0  0.1  0.0         Z
12  16  15   0.0  0.1  0.0         Z
13  16  15   0.0  0.1  0.0         Z
14  16  15   0.0  0.1  0.0         Z
15  16  15   0.0  0.1  0.0         Z
0   0   0   0.0  0.0  0.0
C ***** DATA SET 3: ELEMENT INDICES
224  8  0                    NEL   NMAT   KCP
 1  1   2  17  16  1  14  16
C ***** DATA SET 4: MATERIAL CORRECTION
134                     NCM
1   3  14 3   0
2   3  14 3   0
3   3  14 3   0
6   8  1   2   0
20  8  1   2   0
34  8  1   2   0
48  8  1   2   0
68  5  14 2 0
69  5  14 2 0
70  5  14 2 0
180  3  14 3   0
181  3  14 3   0
182  3  14 3   0
152 2 1   4   0
166 2 1   4   0
```

```
4   3 14 4   0
5   3 14 4   0
57  2 1  6   0
71  2 1  6   0
60  1 1  5   0
74  1 1  5   0
62  3 1  7   0
76  3 1  7   0
66  1 1  8   0
80  1 1  8   0
94  3 14 7   0
95  3 14 7   0
153 1 1  5   0
167 1 1  5   0
181 3 14 6   0
182 3 14 6   0
0   0 0  0   0
C ***** DATA SET 5: MATERIAL PROPERTIES
10   7   1              NMPPM  NCC  IRXN
1.0D-2 1.0D-2 0.0  1.0d3 7.316d10 0.0d-2 0.0d-2 0.00d-5 0.3 1414.8  PROP  MAT 1
0.0  0.0  0.0  0.0  0.0  0.0  0.0      DKD
1.0d-2  1.0d-2  1.0d-2  1.0d-2  1.0d-2  1.0d-2  1.0d-2   TRANC
1.0D0  1.0D0  0.0  1.0d3 7.316d10 0.0d0  0.0d0  0.0D0  0.3  1901.9  PROP  MAT 2
0.0  0.0  0.0  0.0  0.0  0.0  0.0      DKD
0.0D0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   TRANC
1.0D-1 1.0D-1 0.0 1.0d3 7.316d10 0.0D-3 0.0D-3 0.00D-5  0.3  1689.5  PROP  MAT 3
0.0  0.0  0.0  0.0  0.0  0.0  0.0      DKD
1.0d-4  1.0d-4  1.0d-4  1.0d-4  1.0d-4  1.0d-4  1.0d-4   TRANC
3.16D-1 3.16D-1 0.0 1.0D3 7.316d10 0.00D-3 0.00D-3 0.00D-4 0.3 1755.8 PROP MAT 4
0.0  0.0  0.0  0.0  0.0  0.0  0.0      DKD
0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   TRANC
5.62D-2 5.62D-2 0.0 1.0D3 7.316d10 0.00D-3 0.00D-3 0.00D-5 0.3 1472.8 PROP MAT 5
0.0  0.0  0.0  0.0  0.0  0.0  0.0      DKD
0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   TRANC
3.16D-2 3.16D-2 0.0 1.0D3 7.316d10 0.00D-3 0.00D-3 0.00D-5 0.3 1515.8 PROP MAT 6
0.0  0.0  0.0  0.0  0.0  0.0  0.0      DKD
1.0d-3  1.0d-3  1.0d-3  1.0d-3  1.0d-3  1.0d-3  1.0d-3   TRANC
1.0D-1 1.0D-1 0.0 1.0D3 7.316d10 0.00D-3 0.00D-3  0.00D-5 0.3 1512.4 PROP MAT 7
0.0  0.0  0.0  0.0  0.0  0.0  0.0      DKD
0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   TRANC
3.16D-1 3.16D-1 0.0 1.0D3 7.316d10 0.00D-3 0.00D-3 0.00D-4 0.3 1706.1 PROP MAT 8
0.0  0.0  0.0  0.0  0.0  0.0  0.0      DKD
0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   0.0d0   TRANC
   1.0d3  1.0d3  1.0d3  1.0d3  1.0d3  1.0d3  1.0d3  DINTS
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  RHOMU
C ***** DATA SET 6: MATERIAL PROPERTIES
1 2                              KSP  NSPPM
 -1000.0  1000.0
 -1000.0  1000.0
 -1000.0  1000.0
 -1000.0  1000.0
 -1000.0  1000.0
 -1000.0  1000.0
 -1000.0  1000.0
 -1000.0  1000.0
 0.465    0.465
  0.285   0.285
  0.365   0.365
  0.323   0.323
  0.387   0.387
  0.412   0.412
  0.364   0.364
  0.322   0.322
  1.0     1.0
  1.0     1.0
  1.0     1.0
  1.0     1.0
  1.0     1.0
  1.0     1.0
```

```
   1.0      1.0
   1.0      1.0
   0.0      0.0
   0.0      0.0
   0.0      0.0
   0.0      0.0
   0.0      0.0
   0.0      0.0
   0.0      0.0
   0.0      0.0
C ***** DATA SET 7: TIME CONTROL PARAMETERS
80  1                                    NTI,NDTCHG
0.05  0.0  1.0D0 4.0                      DELT,CHNG,DELMAX,TMAX
52 0 0 0 05 00 00 0050   0 0 0500055000 5000      5          5          5
5          5          5          5          5          5          5          5
00 0 0 0 00 00 0000   0 0 0000000000 0000      0          0          0
0          0          0          0          0          0          0          0
1.0d5  2.0d4  1.0D38                      TDTCH
C ***** DATA SET 9: CONTROL PARAMETERS FOR FLOW
20  100   500  1  0  1                    NCYL,NITERH,NPITERH,KSTRH,KSSH,KGRAV
1.0D-3  1.0D-3  1.0 1.0  1.0  0.0          TOLAH,TOLBH,WF,OMEH,OMIH,cnstkr
C ***** DATA SET 10: I.C. FOR FLOW
1  254  1  0.0  0.0  0.0          I.C. OF H
0   0    0  0.0  0.0  0.0
C ***** DATA SET 11: ELEMENTAL SOURCES FOR FLOW
0  0  0  0                                NSELH,NSPRH,NSDPH,KSAIH
C ***** DATA SET 12: WELL SOURCES FOR FLOW
2  2  3  0                                NWNPH,NWPRH,NWDPH,KWAIH
0.0  0.0  1.0d-6  7.5d-3  1.0D38  7.5d-3              TSOSFH,SOSFH
0.0  0.0  1.0d-6  -4.0d-3  1.0D38  -4.0d-3
1  0  0  17  0                    IWTYPH(1,2)
2  0  0 239  0                    IWTYPH(2,2)
0  0  0  0   0
1  1  1  1   1                    IWTYPH(1..2,1)
0  0  0  0   0
C ***** DATA SET 13: VARAIABLE B.C.
0  0  0  0  0                             NRES,NRNP,NRPR,NRDP,KRAIH
C ***** DATA SET 14: DIRICHLET B.C.
34  2  2  0                               NDNPH,NDPRH,NDDPH,KDAIH
0.0  1.0  1.0D38  1.0                      THDBF,HDBF
0.0  0.0  1.0D38  0.0
1  16  1  1  15                   IDTYPH(1..22,2)
18  16  1  15 15
0   0   0  0   0
1  16  1  1  0                    IDTYPH(1..22,1)
18  16  1  2  0
0  0   0  0  0
C ***** DATA SET 15: CAUCHY B.C.
0  0  0  0  0                             NCESH,NCNPH,NCPRH,NCDPH,KCAIH
C ***** DATA SET 16: NEUMANN B.C.
0  0  0  0  0                             NNESH,NNNPH,NNPRH,NNDPH,KNAIH
C ***** DATA SET 17: CONTROL PARAMETERS FOR TRANSPORT
100  500 1  1  2  0                       nitert,npitert,kstrt,ksst,kvi,miconf
1.0d-3  1.0d-3  1.0  0.0  1.0d0  1.0d0  1.0    tolat,tolbt,wt,wv,omet,omit,alphg
1  1 0 5 5 5 5 2 2 1          izoom,idzoom,iepc,nxa,nya,nxd,nzd,nxw,nyw,idetq
0.0005  0.0005   1.0              ADPEPS, ADPARM
C ***** DATA SET 18: I.C. FOR TRANSPORT
1  254 1  1.77D-4 0.0  0.0          IC FOR microbe #1
0   0   0  0.0  0.0  0.0
1  254  1  0.0  0.0  0.0            IC FOR microbe #2
0   0   0  0.0  0.0  0.0
1  254  1  1.77D-4  0.0  0.0        IC FOR microbe #3
0   0   0  0.0  0.0  0.0
1  254  1  5.0D-3  0.0  0.0         IC FOR substrate
0   0   0  0.0  0.0  0.0
1  254  1  5.0D-3  0.0  0.0         IC FOR oxygen
0   0   0  0.0  0.0  0.0
1  254  1  5.0D-3  0.0  0.0         IC FOR nitrate
0   0   0  0.0  0.0  0.0
```

```
1  254  1  3.0D-3  0.0  0.0              IC FOR nutrient
0   0   0  0.0  0.0  0.0
C ***** DATA SET 19: MICROBE-CHEMICAL INTERACTION PARAMETERS
 4.0  0.0   4.0   2.5                    GRATE
 0.4  0.4  0.4   0.17                    YCOEFF
1.8D-2 1.8d-2 1.8D-2 1.8D-2              RTARDS    Kso, Ksn
3.0D-5 3.0d-5 3.0D-5 2.0D-5             RTARDO    Ko, Kn
3.0d-4 3.0d-4 3.0D-4 3.0D-4             RTARDN    Kpo, Kpn
1.0  0.375 1.0   0.375                   SCOEFF    gammao, gamman
0.004 0.002 0.004 0.002                  ECOEFF    alphao, alphan
0.02 0.02 0.02 0.02                      DCOEFF    lambdao, lambdan
3.0D-5 2.0d-5  3.0D-5 2.0D-5             SATURC    GAMMAo, GAMMAn
0.05 0.021 0.05 0.021                    PCOEFF    Epsilon
  1.1d-4                                 COFK
C ***** DATA SET 20: ELEMENTAL SOURCES FOR TRANSPORT
0   0   0   0                            NSEL,NSPR,NSDP,ksai
C ***** DATA SET 21: WELL SOURCES FOR TRANSPORT
2   2   3   0                            NWNPH,NWPRH,NWDPH,KWAIH
0.0  0.0  0.0  1.0d-6  7.5d-3  0.0  1.0D38  7.5d-3 0.0
0.0  0.0  0.0  1.0d-6 -4.0d-3 0.0  1.0D38  -4.0d-3 0.0
  17  239
1  1   1   1    1
0  0   0   0    0
1  1   1   1    1
0  0   0   0    0
1  1   1   1    1
0  0   0   0    0
1  1   1   1    1
0  0   0   0    0
1  1   1   1    1
0  0   0   0    0
1  1   1   1    1
0  0   0   0    0
1  1   1   1    1
C ***** DATA SET 22: VARIABLE B.C. FOR TRANSPORT
 16  17  1  2  0                         NVES,NVNP,NVPR,NVDP,kdai
 0.0  0.0  1.0d38  0.0d0                  Profile
 1  15  1  1  0                          Profile type for microbe #1
 0  0   0  0  0
 1  15  1  1  0                          Profile type for microbe #2
 0  0   0  0  0
 1  15  1  1  0                          Profile type for microbe #3
 0  0   0  0  0
 1  15  1  1  0                          Profile type for substrate
 0  0   0  0  0
 1  15  1  1  0                          Profile type for oxygen
 0  0   0  0  0
 1  15  1  1  0                          Profile type for nitrate
 0  0   0  0  0
 1  15  1  1  0                          Profile type for nutrient
 0  0   0  0  0
 1  16  1  15 15
 0  0   0  0  0                          NPDB
 1  15  14 1  2  1  14  1  1             VB element side
 0  0   0  0  0  0   0   0
C ***** DATA SET 23: DIRICHLET B.C. FOR TRANSPORT
17  5  2  0                              DIRICHLET BOUNDARY CONDITION
0.0  1.77D-4  1.0D38  1.77D-4            TCDBF  CDBF    FOR microbe #1
0.0  0.0      1.0D38  0.0
0.0  1.5D-2   1.0D38  1.5D-2                              FOR substrate
0.0  5.0D-3 1.0D38  5.0D-3                                FOR oxygen
0.0  3.0D-3 1.0D38  3.0D-3                                FOR nutrient
1  16  1  1  0                           IDTYP for microbe #1
0  0   0  0  0
1  16  1  2  0                           IDTYP for microbe #2
0  0   0  0  0
1  16  1  1  0                           IDTYP for microbe #3
0  0   0  0  0
```

```
1   16  1   3   0                          IDTYP for substrate
0    0   0   0   0
1   16  1   4   0                          IDTYP for oxygen
0    0   0   0   0
1   16  1   4   0                          IDTYP for nitrate
0    0   0   0   0
1   16  1   5   0                          IDTYP for nutrient
0    0   0   0   0
1   16   1   1   15                                NPDB
0    0   0   0    0
C ***** DATA SET 24: CAUCHY B.C. FOR TRANSPORT
0   0  0  0   0                                nces,ncnp,ncpr,ncdp,kcai
C ***** DATA SET 25: NEUMANN B.C. FOR TRANSPORT
0   0  0  0   0                                nnes,nnnp,nnpr,nndp,knai
0
```

Simulation Results

The results of velocity fields and concentration contours simulated at t = 2 days and 4 days are

shown in Figure 4.5 and Figure 4.6.
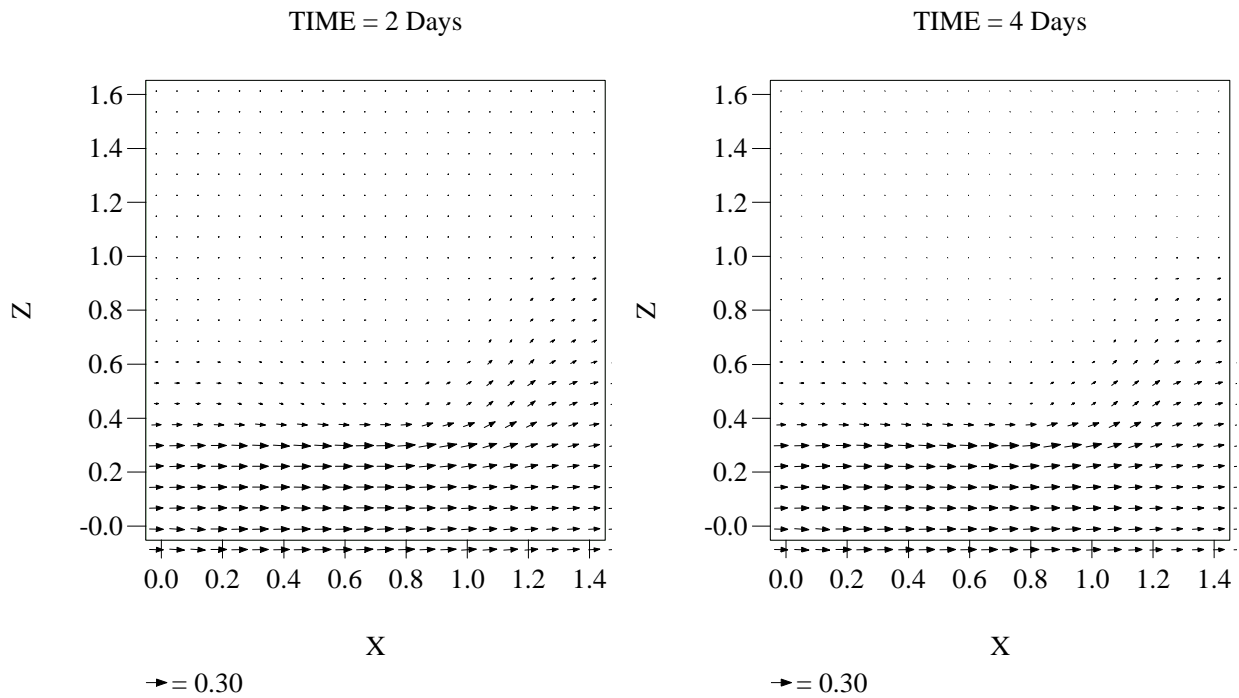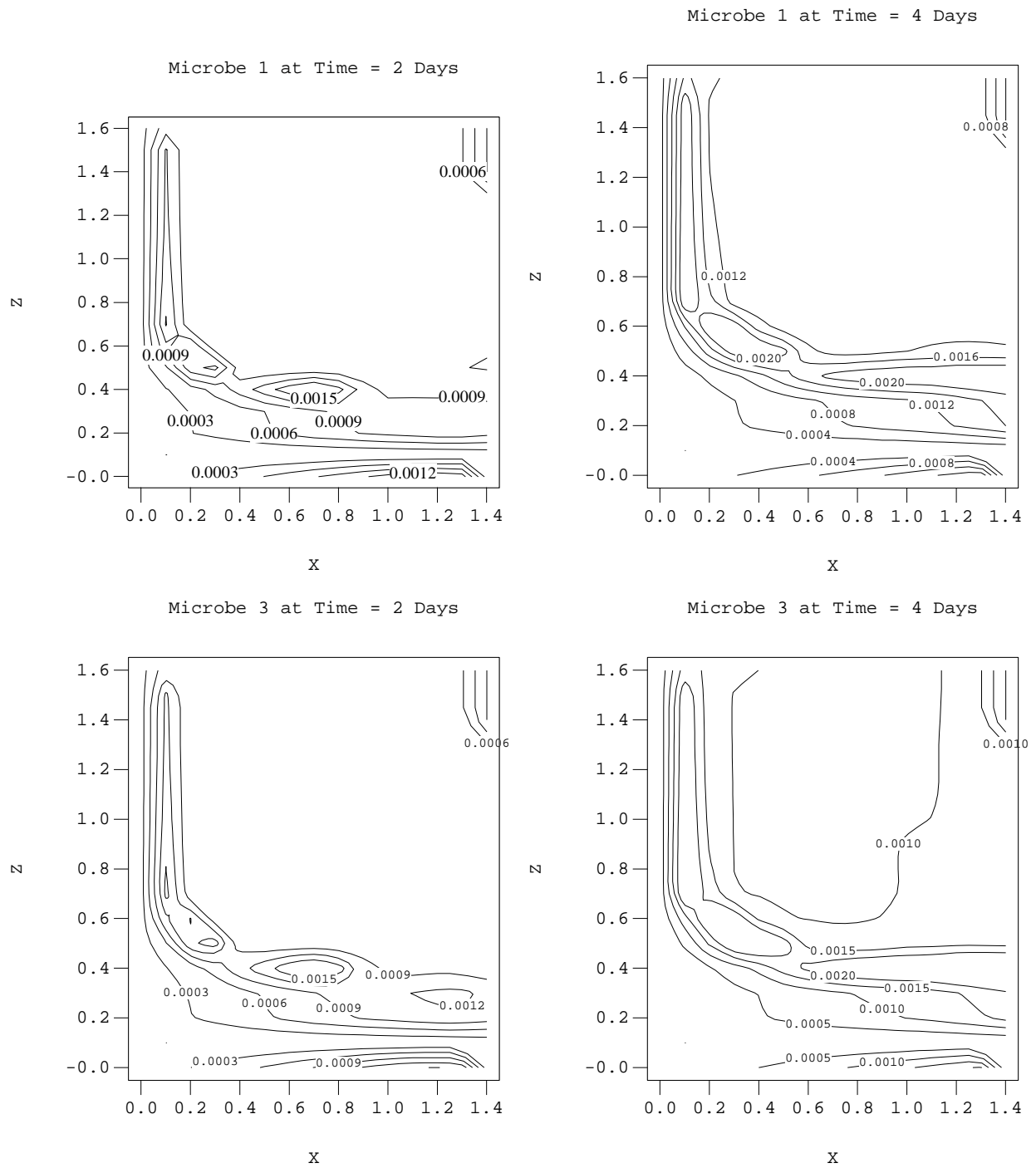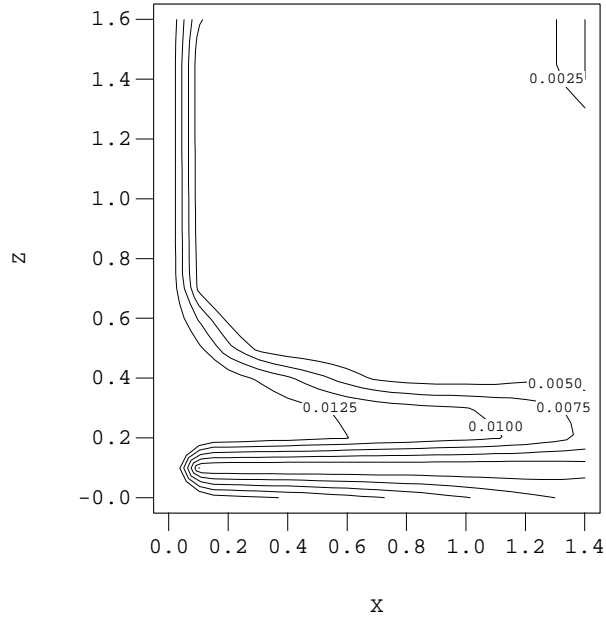

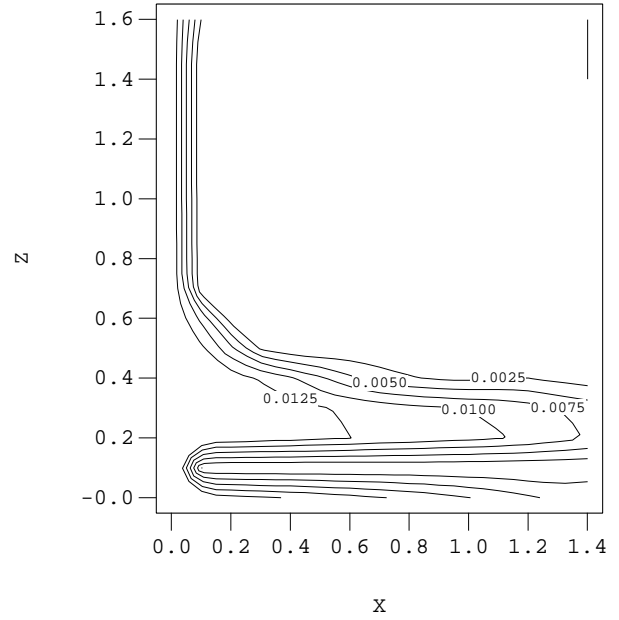Figure 4.5 The flow field at (a) t = 2 days and (b) t = 4 days

The following three pages are Figure 4.6. They contain the simulation results of transport and fate of chemicals and microbes.

Microbe 1 at Time = 4 Days

Microbe 1 at Time = 2 Days

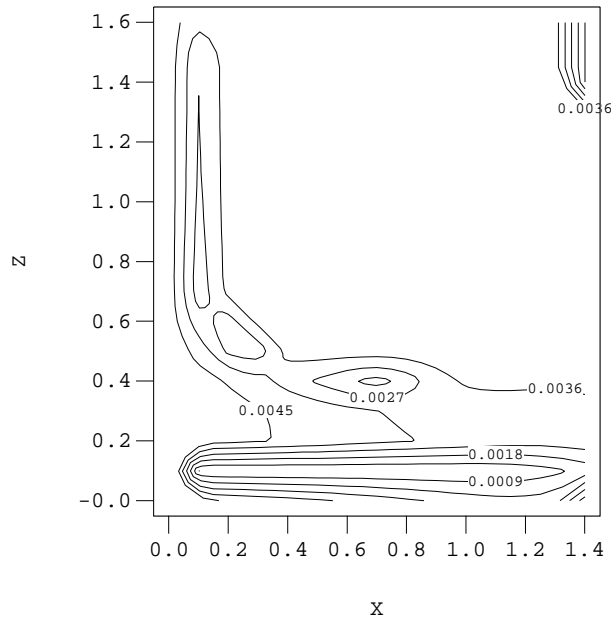Microbe 3 at Time = 2 Days

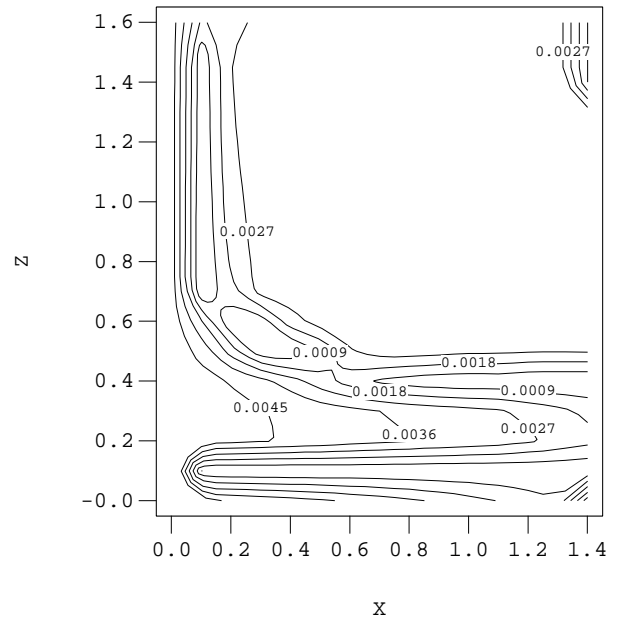Microbe 3 at Time = 4 Days

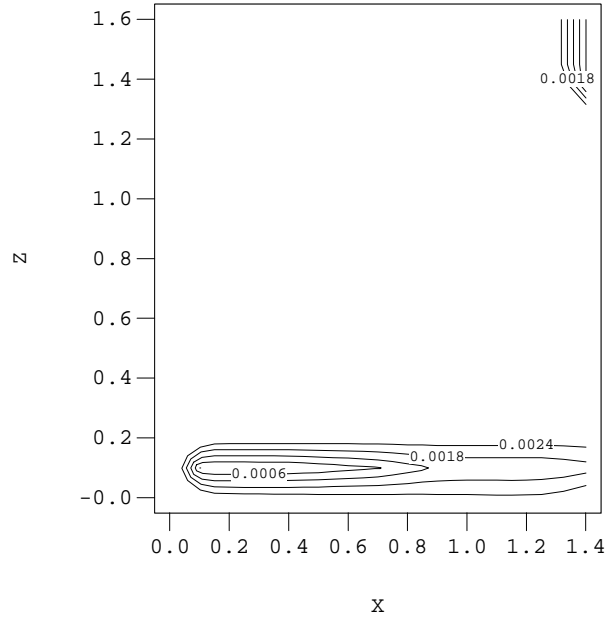Substrate at Time = 2 Days
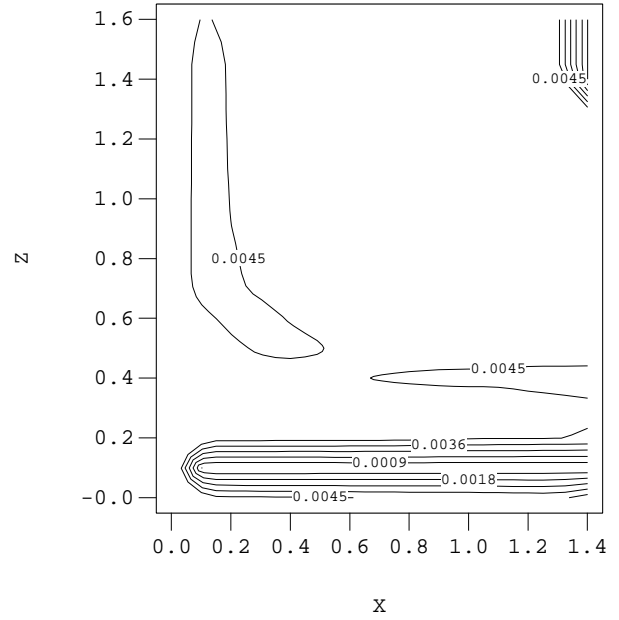
Substrate at Time = 4 Days

Oxygen at Time = 2 Days
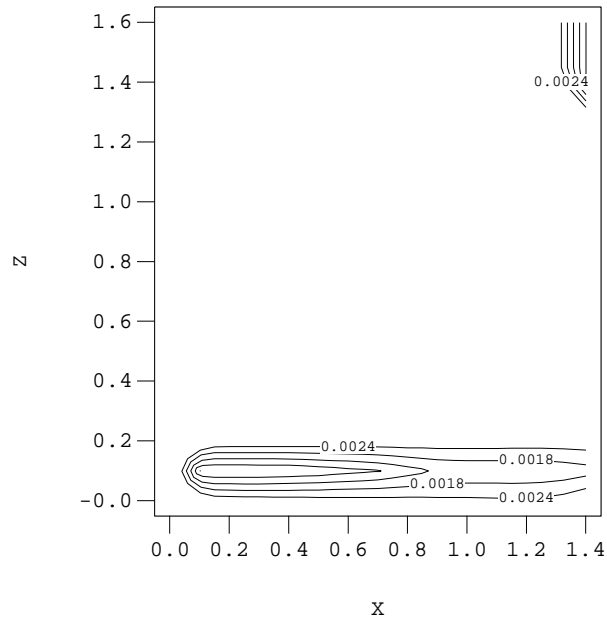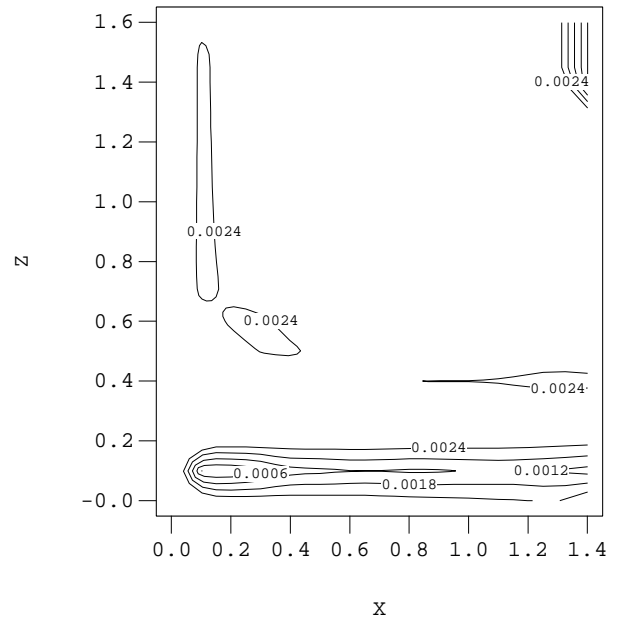
Oxygen at Time = 4 Days

Nitrate at Time = 2 Days

Nitrate at Time = 4 Days

Nutrient at Time = 2 Days

Nutrient at Time = 4 Days

APPENDIX A: Data Input Guide

**** Data sets 2 through 26 must be preceded by a card *****

******** containing the description of the data set ********

1. PROBLEM IDENTIFICATION AND DESCRIPTION
    Two Records per Problem: FORMAT(I5,A72)

    Record one contains the following variables.
    (1) NPROB = Problem number
    (2) TITLE = Array for the title of the problem. It may contain up to 72 characters from column 6 to
        column 77.

    Record two contains the following 8 variables (FREE FORMAT)
    (1) IITR = Is non-linear convergence table information to be printed? 0= No, 1 = Yes.
    (2) ICHNG = Is cyclic change of rainfall-seepage nodes to be printed? 0 = No, 1 = Yes.
    (3) INTER = Is non-linear iterate information to be printed? 0 = No, 1 = Yes.
    (4) IBUG = Is debug information to be printed? 0 = No, 1 = Yes.
    (5) IMOD = Indicator of simulated problems:
        0 = Read input data only,
        1 = Only transport is simulated,
        10 = Only flow is simulated,
        11 = Both flow and transport are simulated.
    (6) NITRHT = No. of iterations allowed for solving steady-state flow-transport iteration.
    (7) OMEHT = Iteration parameter of flow for coupled flow-transport iteration loop:
        $0.0 \sim 1.0$ = Under-relaxation,
        1.0 = Exact relaxation,
        $1.0 \sim 2.0$ = Over-relaxation.
    (8) IGEOM = Integer indicating if (1) the geometry, boundary and pointer arrays are to be printed;
        (2) the boundary and pointer arrays are to be computed or read via logical units. If to be
        computed, they should be written on logical units. If IGEOM is even number, (1) will not be
        printed. If IGEOM is odd number, (1) will be printed. If IGEOM is less than or equal to 1,
        boundary arrays will be computed and written on logical unit, but if IGEOM is greater than
        1, boundary arrays will be read via logical unit 13. If IGEOM is less than or equal to 3,
        pointer arrays will be will be computed and written on logical unit 14, but if IGEOM is
        greater than 3, pointer arrays will be read via logical unit 14.

2. OPTION PARAMETERS FOR BOTH FLOW AND TRANSPORT
    One record contains the following 10 variables (FREE FORMAT)
    (1) IMID = Mid-difference control: 0 = No mid-difference, 1 = Mid-difference is used. (W=1.0)
    (2) IWET = Weighting factor control: 0 = Galerkin weighting, 1 = Upstream weighting.

(3) ILUMP = Mass matrix lumping control: 0 = No lump, 1 = Lump.
(4) IOPTIM = Optimization factor for computing indicator: 0 = Optimization factor is set to be 1.0, 0.0, -1.0, or APHAG by the program depending on the velocity and dispersion coefficient, 1 = Optimization factor is to be computed.
(5) IPNTS = Is pointwise iteration method to be used to solve the matrix equations? 0 = No, 1 =Yes.
(6) IVML = Is velocity matrix to be lumped? 0 = No, 1 = Yes.
(7) NSTRH = No. of logical records to be read via logical unit 11 for restarting calculation. 0 = No restart.
(8) NSTRT = No. of logical records to be read via logical unit 12 for restarting calculation. 0 = No restart.
(9) IQUAR = Index of using quadrature for numerical integration;
  11 = Nodal quadrature for surface integration, Nodal quadrature for element integration,
  12 = Nodal quadrature for surface integration, Gaussian quadrature for element integration,
  21 = Gaussian quadrature for surface integration, Nodal quadrature for element integration.
  22 = Gaussian quadrature for surface integration, Gaussian quadrature for element integration,
(10) IDISP = Is dispersion/diffusion to be included in the transport simulation? 0 = No, 1 = yes.

3.   NODAL POINT COORDINATE
This data set is required for both NSTRH and NSTRT equal to zero. Two subsets are included. Subset 1 contains one record telling the program total number of nodes in the region of interest. Subset 2 usually requires 2*NNP records, NNP record for the x-coordinate and NNP record for the z-coordinate. However, if a group of subsequent nodes appear in regular pattern, automatic generation can be made.

A. Subset 1 - FREE FORMAT: it contains one variable.
(1) NNP = total number of global nodes in the region of interest.

B. Subset 2 - FREE FORMAT : Each record contains the following variables.
(1) NI = Node number of the first node in the sequence.
(2) NSEQ = NSEQ subsequent nodes will be automatically generated.
(3) NAD = Increment of node number for each of the NSEQ subsequent nodes.
(4) XNI = x-coordinate of node NI, (L).
(5) XAD = Increment of x-coordinate for each of the NSEQ subsequent nodes, (L).
(6) XRD = Percentage of the increase of the increment over its preceding increment, (Decimal point); If XRD.EQ.0., all increments XAD's are the same. If XRD .GT. 0.0, the first increment is XAD*(1+XRD), the second increment is XAD*(1+XRD)**2, the third increment is XAD*(1+XRD)**3, and so on.

   *** NOTE: A line with 6 0's must be used to signal the end of this data set.

***** A similar group of records is used to input z-coordinate *****

4.   ELEMENT INCIDENCE
This data set is required for both NSTRH and NSTRT equal to zero. Two subsets are included.

Subset 1 contains three variables telling the program the total number of global elements and material types in the region of interest, and the control index of material property. Subset 2 usually requires NEL records. However, if a group of elements appear in a regular pattern, automatic generation is made.

A. Subset 1 - FREE FORMAT: it contains three variables.
(1) NEL = total number of elements in the region of interest.
(2) NMAT = No. of materials.
(3) KCP = Permeability input control:
       0 = Input hydraulic saturated conductivity,
       1 = Input intrinsic saturated permeability.

B. Subset 2 - FREE FORMAT: Each record contains the following variables.
(1) MI = Global element number.
(2) IE(MI,1) = Global node number of the first node of element MI.
(3) IE(MI,2) = Global node number of the second node of element MI.
(4) IE(MI,3) = Global node number of the third node of element MI.
(5) IE(MI,4) = Global node number of the forth node number MI.
(6) IE(MI,5) = Material type to be applied to element MI.
(7) MODL = Number of elements in the direction of most rapidly numbered nodes.
(8) NLAY = Number of elements in the direction of least rapidly numbered nodes.

IE(MI,1) IE(MI,4) are numbered beginning with the lower left corner and progressing around the element in a counterclockwise direction. For a rectangular block of elements, it is only necessary to specify the first element, the width MODL and the length NLAY, where MODL and NLAY are measured in elements. The following figure provides an example. The object is considered to be rectangular since it has width MODL = 3 on two opposite sides and length NLAY = 5 on the other two opposite sides. To generate definitions of elements 2 through 15 automatically, including both the incidence and material type, only one card is necessary: Although all elements of this example will be assumed to contain the same material type, MTYP = 1, this situation can easily be changed by using material-correction facility.

```
    ----------------------------
    1  1  5  6  2  3  5
    ----------------------------
```

5.   MATERIAL TYPE CORRECTION
     Two subsets of free-formatted data records are required for this data set.

     A. subset 1:
     (1) NCM = Number of elements with material corrections.

     B. subset 2: This set of data records is required only if NCM > 0. Normally, NCM records are
             required. However, if a group of elements appear in a regular pattern, automatic generation
             may be made. Each record contains the following variables.
     (1) MI = Global element number of the first element in the sequence.
     (2) NSEQ = NSEQ subsequent elements will be generated automatically.
     (3) MAD = Increment of element number for each of the NSEQ subsequent elements.
     (4) MITYP = Type of material correction for element MI.
     (5) MTYPAD = Increment of MITYP for each of the NSEQ subsequent elements.

     **** NOTE: A line with 5 0's must be used to signal the end of this data set.

6.   MATERIAL PROPERTIES
     Five subsets of free-formatted data records are required for this data set. There are 3*NMAT+2
     records in this data set.

     A. Subset 1 - FREE FORMAT: it contains three variables.
     (1) NMPPM = No. of parameters used for describing material properties. (It is set to be 10 originally).
     (2) NCC = No. of components in the system. Since the kinetic reaction model is built in the program
             according to Eq. (2.9) through Eq. (2.15), NCC is assigned to 7 and IRXN is set to 1 if the
             microbial-chemical Monod type reactions are involved. When NCC = 7 and the Monod
             kinetics are used, the first component must be Microbe # 1, the second component Microbe
             # 2, the third component Microbe #3, the fourth component the substrate, the fifth component
             Oxygen, the sixth component Nitrate, and the seventh component the nutrient.
             NCC can be 1 for the single component simulation and NCC is equal to 2 by using
             stochiometric model which results in IRXN = -1. NCC can be any value if users modify the
             kinetic model in the program (Subroutine ADVRX).
     (3) IRXN = the index indicating the chemical-microbial kinetic reaction type. -1 refers to
             stochiometric reaction; 1 indicates Monod type reaction.

     **** Subsets 2, 3, and 4 should be repeated NMAT times.
     B. Subset 2 - FREE FORMAT: it contains the following NMPPM variables for each material type.
     (1) PROP(I,1) = Saturated longitudinal hydraulic conductivity (intrinsic permeability) in X-direction
             of the I-th material (L/T or L^2).
     (2) PROP(I,2) = Saturated longitudinal hydraulic conductivity (intrinsic permeability) in Z-direction
             of the I-th material (L/T or L^2).
     (3) PROP(I,3) = Saturated transverse hydraulic conductivity (intrinsic permeability) of the I-th

material (L/T or L^2).
(4) PROP(I,4) = Density of groundwater in the I-th material (M/L^3).
(5) PROP(I,5) = Gravity in the I-th material (L/T^2).
(6) PROP(I,6) = Longitudinal dispersivity in the I-th material (L).
(7) PROP(I,7) = Transverse dispersivity in the I-th material (L).
(8) PROP(I,8) = Diffusion coefficient in the I-th material (L^2/T).
(9) PROP(I,9) = Viscosity of groundwater in the I-th material (M/LT).
(10) PROP(I,10) = Bulk density of the I-th material (M/L^3).

.
.
.

(NMPPM) PROP(I,NMPPM) = The NMPPM-th material parameter of the I-th material.

C. Subset 3 - FREE FORMAT: Each record contains NCC parameters (L^3/M) as follows.
(1) RKD(I,1) = Distribution coefficient of Microbe #1 in the I-th material.

.
.
.

(NCC) RKD(I,NCC) = Distribution coefficient of the NCC-th component in the I-th material.

D. Subset 4 - FREE FORMAT: each record contains NCC parameters (1/T) as follows.
(1) TRANC(I,1) = Transformation rate constants of Microbe #1 in the I-th material.

.
.
.

(NCC) TRANC(I,NCC) = Transformation rate constants of the NCC-th component in the I-th material.

E. Subset 5 - one record of FREE FORMAT: it contains NCC parameters (M/L^3) as follows.
(1) DINTS(1) = Intrinsic density of Microbe #1.

.
.
.

(NCC) DINTS(NCC) = Intrinsic density of the NCC-th component.

F. Subset 6 - one record of FREE FORMAT: it contains NCC parameters (L^2/T) as follows.
(1) RHOMU(1) = Viscosity effecting factor of Microbe #1.

.
.
.

(NCC) RHOMU(NCC) = Viscosity effecting factor of the NCC-th component.

7.   SOIL PROPERTIES
      Three or five subsets of free-formatted data records are required for this data set depending on the given forms of the soil property functions.

      A. Subset 1: Soil property control parameters

(1) KSP = Soil property input control, 0 = analytical input, 1 = Tabular data input.
(2) NSPPM = Number of points in tabular soil property functions or number of parameters to specify analytical soil functions per material.

B. Subset 2a: Analytical soil parameters - This sub-data-set is needed if and only if KSP is 0. Two sets of records are required, one for moisture-content parameters and the other for conductivity (permeability) parameters.
(1) SPP(J,I,1) = Analytical moisture-content parameter J of material I, J = 1..NSPPM. NMAT sets of these parameters are required for I = 1..NMAT. That is, if SPP(J,I,1) for J = 1..NSPPM can be put on a single line, we need NMAT consecutive line for the sets of parameters.
(2) SPP(J,I,2) = Analytical relative conductivity parameter J of material I. Similar input data setting is required for these parameters as for SPP(J,I,1)

C. Subset 2b: Soil properties in tabular form - This sub-data-set is needed if and only if KSP not is 0. Four sets of records are needed -- for pressure, water-content, relative conductivity (or relative permeability), and water capacity, respectively.
(1) SPP(J,I,4) = Tabular value of pressure head of the J-th point for material I. NMAT sets of these parameters are required for I = 1..NMAT. That is, if SPP(J,I,4) for J = 1..NSPPM can be put on a single line, we need NMAT consecutive line for the sets of parameters.
(2) SPP(J,I,1) = Tabular value of moisture-content of the J-th point in material I. Similar input data setting is required for these parameters as for SPP(J,I,4)
(3) SPP(J,I,2) = Tabular value of relative conductivity of the J-th point in material I. Similar input data setting is required for these parameters as for SPP(J,I,4)
(4) SPP(J,I,3) = Tabular value of moisture-content capacity of the J-th point in material I. Similar input data setting is required for these parameters as for SPP(J,I,4)

8.   TIME CONTROL PARAMETERS
Five subsets of data records are required for this data set.

A. Subset 1: free format
(1) NTI = Number of time steps or time increments.
(2) NDTCHG = No. of times to reset time step size to initial time step size.

B. Subset 2: free format
(1) DELT = Initial time step size, (T).
(2) CHNG = Percentage of change in time step size in each of the subsequent time increment, (dimensionless in decimal point).
(3) DELMAX = Maximum value of DELT, (T).
(4) TMAX = Maximum simulation time, (T).

C. Subset 3: format = 80I1
(1) KPR0 = Printer control for steady state and initial conditions;
        0 = print nothing,
        1 = print FLOW, FRATE, and TFLOW,
        2 = print above (1) plus pressure head H,
        3 = print above (2) plus total head,
        4 = print above (3) plus moisture content,

5 = print above (4) plus Darcy velocity.
(2) KPR(I) = Printer control for the I-th (I = 1, 2, ..., NTI) time step similar to KPR0.

D. Subset 4: format = 80I1
(1) KDSK0 = Auxiliary storage control for steady state and initial condition;
     0 = no storage, 1 = store on Logical Unit 1.
(2) KDSK(I) = Auxiliary storage control for the I-th (I = 1, 2, ..., NTI) time step similar to KDSK0.

E. Subset 5: free format
(1) TDTCH(I) = Time when the I-th (I = 1, 2, ..., NDTCHG) step-size-resetting is needed.

**** NOTE: NTI can be computed by NTI = I1 + 1 + I2 + 1, where I1 = largest integer not exceeding Log(DELMAX/DELT)/Log(1+CHNG),   I2 = largest integer not exceeding (RTIME-DELT*((1+CHNG)**(I1+1)-1)/CHNG)/DELMAX,   RTIME = Real simulation time, DELMAX,DELT,and CHNG are defined in Data Set 3.

The following data sets, data set 9 to data set 16, must be skipped as IMOD equals 1.

9.   BASIC REAL AND INTEGER PARAMETERS FOR FLOW
      Two subsets of free-formatted data records are required for this data set.

A. Subset 1 - one record of basic integer for flow part (FREE FORMAT).
(1) NCYL = No. of cycles allowed for iterating variable boundary conditions.
(2) NITERH = No. of iterations allowed for solving nonlinear equations.
(3) NPITERH = No. of iterations allowed for pointwise iteration.
(4) KSTRH = Auxiliary storage output control: 0 = No storage, 1 = Output stored on logical unit 11.
(5) KSSH = Steady state control for flow: 0 = Steady state solution is desired, 1 = Only transient
     solutions are desired.
(6) KGRAV = Gravity term control: 0 = No gravity term (Used for horizontal flow only),
     1 = With gravity term.

B. Subset 2 - one record of basic real number for flow part (FREE FORMAT).
(1) TOLAH = Steady-state convergence criteria,
(2) TOLBH = Transient-state convergence criteria.
(3) WF = Time derivative weighting factor: 0.5 = Crank-Nicolson central, 1.0 = Backward difference
     and/or mid-difference.
(4) OMEH = Iteration parameter for solving the nonlinear matrix equation:
     0.0 ~ 1.0 = Under-relaxation,
     1.0 = Exact relaxation,
     1.0 ~ 2.0 = Over-relaxation.
(5) OMIH = Relaxation parameter for solving the matrix equation pointwisely:
     0.0 ~ 1.0 = Under-relaxation,
     1.0 = Exact relaxation,
     1.0 ~ 2.0 = Over-relaxation.
(6) CNSTKR = Constraint on relative hydraulic conductivity;
     0 = no constraint,

111

0.0001, 0.001, or 0.01 should be tried when nonconvergency occurs in solving the nonlinear flow equation.

10. CARD INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS OF FLOW PART
Free-formatted data records are required for this data set. Generally, NNP records are needed. However, if a group of nodes appear in a regular pattern, auto-generation is made.

Initial pressure head - each record contains the following variables.
(1) NI = Global node number of the first node in the sequence.
(2) NSEQ = NSEQ subsequent nodes will be generated automatically.
(3) NAD = Increment of node number for each of the NSEQ nodes.
(4) HNI = Initial or pre-initial pressure head of node NI, (L).
(5) HAD = Increment of initial or pre-initial head for each of the NSEQ nodes, (L).
(6) HRD = 0.0

**** NOTE: A line with 6 0's must be used to signal the end of this data set.

NOTE ON INITIAL CONDITIONS AND RESTARTING: The initial condition for a transient calculation may be obtained in two different ways: from card input, or steady-state calculation using a time-invariant boundary condition that is different from those for transient computation. In the latter case a card input of the pre-initial conditions is required as the zero-th order iterate of the steady state solution.

NOTE ON STEADY STATE INPUT: Steady state option may be used to provide either the final state of a system under study or the initial conditions for a transient state calculation. In former case KSSh = 0, KSSt $\geq$ 0, and NTI = 0, and in the latter case KSSh = 0 or KSSt $\geq$ 0 and NTI > 0. If KSSh > 0, and KSSt > 0, there will be no steady state calculation.

11. ELEMENT (DISTRIBUTED) SOURCE/SINK FOR FLOW SIMULATIONS
Four subsets of free-formatted data records are required in this data set.

A. Subset 1: control parameters
(1) NSELH= No. of source/sink elements.
(2) NSPRH= No. of source/sink profiles.
(3) NSDPH= No. of data points in each of the NSPR source/sink profiles.
(4) KSAIH= Is element-source/sink profile to be input analytically, 0 = no, 1 = yes.

B. Subset 2: source/sink profiles - This group of data is needed if and only if NSELH .GT. 0. For each sub-data-record, NSDPH of the data pair (TSOSFH(J,I),SOSFH(J,I)) are required. If this sub-data-record can be fitted in a line, we will need NSPRH lines.
(1) TSOSFH(J,I) = Time of the J-th data point in the I-th profile, (T).
(2) SOSFH(1,I) = Source/sink value of the J-th data point in the I-th profile, (L**3/T/L**2/L).

C. Subset 3: global source/sink element number - This group of data is needed if and only if NSELH .GT. 0. NSELH data points are required for this record.
(1) MI = The first compressed element in the sequence.

112

(2) NSEQ = NSEQ elements will be generated automatically.
(3) MAD = Increment of element number for each of the NSEQ elements.
(4) MITYP = Global element number of element MI.
(5) MTYPAD = Increment of MITYP for each of the NSEQ elements.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

D. Subset 4: Source type assigned to each element - Usually one record per element. However,
     automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following.
(1) MI = Global element number of the first element in the sequence.
(2) NSEQ = NSEQ elements will be generated automatically.
(3) MAD = Increment of element number for each of the NSEQ elements.
(4) MITYP = Source type in element MI.
(5) MTYPAD = Increment of MITYP for each of the NSEQ elements.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

12. POINT (WELL) SOURCE/SINK DATA FOR FLOW SIMULATION
     Four subsets of free-formatted data records are required for this data set.

A. Subset 1: control parameters
(1) NWNPH = No. of well or point source/sink nodal points.
(2) NWPRH = No. of well or point source/sink strength profiles.
(3) NWDPH = No. of data points in each of the NWPR profiles.
(4) KWAIH = Is well-source/sink profile to be input analytically, 0 = no, 1 = yes.

B. Subset 2: source/sink profiles - This group of data is needed if and only if NWNPH .GT. 0. For
     each sub-data-record, NWDPH of the data pair (TWSSFH(J,I),WSSFH(J,I)) are required.
     If this sub-data-record can be fitted in a line, we will need NWPRH lines.
(1) TWSSFH(J,I) = Time of the J-th data point in the I-th profile, (T).
(2). WSSFH(J,I) = Source/sink value of the J-th data point in the I-th profile, $(L**3/T/L)$.

C. Subset 3: global source/sink node number - This group of data is needed if and only if NWNPH
     .GT. 0. NWNPH data points are required for this record.
(1) NI = Compressed well node number of the first node in the sequence.
(2) NSEQ = NSEQ nodes will be generated automatically.
(3) NAD = Increment of well node number for each of the NSEQ nodes.
(4) NITYP = Global node number of node NI.
(5) NTYPAD = Increment of NITYP for each of the NSEQ nodes.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

D. Subset 4: Source type assigned to each well - Usually one record per element. However, automatic
     generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following.
(1) NI = Compressed well node number of the first node in the sequence.
(2) NSEQ = NSEQ nodes will be generated automatically.

(3) NAD = Increment of well node number for each of the NSEQ nodes.
(4) NITYP = Source type in node NI.
(5) NTYPAD = Increment of NITYP for each of the NSEQ nodes.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

13. RAINFALL/EVAPORATION-SEEPAGE BOUNDARY CONDITIONS
   Seven subsets of data records are required for this data set.

   A. Subset 1: control parameters
   (1) NRES = No. of variable boundary element sides.
   (2) NRNP = No. of variable boundary nodal points.
   (3) NRPR = No. of rainfall profiles,
   (4) NRDP = No. of rainfall data points in each of the NRPR rainfall profiles.
   (5) KRAIH = Is rainfall profile to be input analytically, 0 = no, 1 = yes.

   B. Subset 2: boundary profiles - This subset is required only when NRES is not 0. NRPR profiles are
         needed. For each profile, NRDP of the data pair (TRF(J,I),RF(J,I)) are required. If these data
         pairs can fit in a line, we will need NRPR of data lines.
   (1) TRF(J,I) = Time of the J-th data point in the I-th profile, (T).
   (2) RF(J,I) = Rainfall/evaporation rate of the J-th data point in the I-th profile, (L/T).

   C. Subset 3: Global Node Number of All Compressed Variable Boundary (VB) Nodes. At most,
         NRNP records are needed for this subdata set, one each for NRNP variable boundary nodes.
         For I-th (I = 1, 2, ...,) Record, it contains the following 5 variables.
   (1) NI = Compressed VB node number of the first node in the sequence.
   (2) NSEQ = NSEQ nodes will be generated automatically.
   (3) NIAD = Increment of NI for each of the NSEQ nodes.
   (4) NODE = Global node number of node NI,
   (5) NODEAD = Increment of NODE for each of the NSEQ nodes.

   **** NOTE: A line with 5 0's is used to signal the end of this data set.

   D. Subset 4: boundary profile types assigned to each VB point. At most NRNP records are needed.
         However, automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the
         following variables.
   (1) MI = Compressed VB node of the first node in the sequence.
   (2) NSEQ = NSEQ nodes will be generated automatically.
   (3) MIAD = Increment of NI for each of the NSEQ nodes.
   (4) MITYP = Type of rainfall/evaporation profiles assigned to node MI.
   (5) MTYPAD = Increment of MITYP for each of the NSEQ nodes.

   **** NOTE: A line with 5 0's is used to signal the end of this data set.

   E. Subset 5: Ponding Depth Allowed in Each of NRNP Variable Boundary Nodes. Normally, NRNP
         records are needed, one for each of the NRNP nodes. However, if a group of nodes has a

114

regular pattern of ponding depth, automatic generation is made. For I-th (I = 1, 2, ...,) record, it contains the following variables.

(1) NI = Compressed VB node number of the first node in a sequence.
(2) NSEQ = NSEQ subsequent nodes will be generated automatically.
(3) NIAD = Increment of NI for each of the NSEQ subsequent nodes.
(4) HCONNI = Ponding depth of node NI, (L).
(5) HCONAD = Increment of HCONNI for each of the NSEQ nodes, (L).
(6) HCONRD = 0.0

**** NOTE: A line with 6 0's must be used to signal the end of this subdata set.

F. Subset 6: Minimum Pressure Head Allowed in Each NRNP Variable Boundary Nodes. This subdata set is read-in similar to the above subdata set. For I-th (I = 1, 2, ...,) record, it contains the following variables.
(1) NI = Compressed VB node number of the first node in a sequence.
(2) NSEQ = NSEQ subsequent nodes will be generated automatically.
(3) NIAD = Increment of NI for each of the NSEQ subsequent nodes.
(4) HMINNI = Minimum pressure head allow for node NI, (L).
(5) HMINAD = Increment of HMINNI for each of the NSEQ nodes, (L).
(6) HMINRD = 0.0

**** NOTE: A line with 6 0's must be used to signal the end of this subdata set.

G. Subset 7: Specification of Rainfall/evaporation-seepage Sides. Normally, NRES records are required, one each for a variable boundary (VB) element side. However, if a group of rainfall/evaporation-seepage element sides appears in a regular pattern, automatic generation may be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
(1) MI = Compressed VB element side number of the first element side in a sequence.
(2) NSEQ = NSEQ subsequent VB element sides will be generated automatically.
(3) M = Global element no. to which the MI-th RS side belongs.
(4) IS1 = Compressed RS node number of the first node of element side MI.
(5) IS2 = Compressed RS node no. of the 2-nd node of the MI-th RS side.
(6) MIAD = Increment of MI for each of the NSEQ subsequent VB element sides.
(7) MAD = Increment of M for each of the NSEQ subsequent RS side.
(8) IS1AD = Increment of IS1 for each of the NSEQ subsequent RS side.
(9) IS2AD = Increment of IS2 for each of the NSEQ subsequent RS side.

**** NOTE: A line with 9 0's must be used to signal the end of this subdata set.

14. DIRICHLET BOUNDARY CONDITIONS FOR FLOW SIMULATION
Four subsets of data records are required for this data set.

A. Subset 1: control parameters
(1) NDNPH = No. of Dirichlet nodal points, should be .GE. 1.
(2) NDPRH = No. of total Dirichlet-head profiles, should be .GE. 1.

(3) NDDPH = No. of data points in each total head profiles, should be .GE. 1.
(4) KDAIH = Is Dirichlet boundary value profile to be input analytically, 0 = no, 1 = yes

B. Subset 2: Dirichlet-head profiles - This sub-data-set is required only if NDNPH is not 0. NDPRH
of profiles are needed. For each profile, NDDPH of the data pair (THDBF(J,I),HDBF(J,I))
are needed. If these data pairs can fit in a line, we will need NDPRH lines.
(1) THDBF(J,I) = Time of the J-th data point in the I-th profile, (T).
(2) HDBF(J,I) = Total head of the J-th data point in the I-th profile, (L).

C. Subset 3: Global Node Number of All Compressed Dirichlet Boundary Nodes. At most, NDNPH
records are needed for this subdata set, one each for NDNPH Dirichlet boundary nodes. For
I-th (I = 1, 2, ...,) record, it contains the following 5 variables.
(1) NI = Compressed Dirichlet node number of the first node in the sequence.
(2) NSEQ = NSEQ subsequent Dirichlet nodes will be generated automatically.
(3) NAD = Increment of NI for each of the NSEQ nodes.
(4) NITYP = Global node number for node NI and NSEQ subsequent nodes.
(5) NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.

**** NOTE: A line with 5 0's must be used to signal the end of this sub-data set.

D. Subset 4: boundary profile type assign to each Dirichlet node - Normally one record per Dirichlet
node, i. e. a total of NDNPH records. However, if the Dirichlet nodes appear in regular
pattern, automatic generation may be made. For I-th (I = 1, 2, ...,) record, it contains the
following variables.
(1) NI = Compressed Dirichlet node number of the first node in the sequence.
(2) NSEQ = NSEQ subsequent Dirichlet nodes will be generated automatically.
(3) NAD = Increment of NI for each of the NSEQ nodes.
(4) NITYP = Type of total head profile for node NI and NSEQ subsequent nodes.
(5) NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.

**** NOTE: A line with 5 0's must be used to signal the end of this sub-data set.

15. CAUCHY BOUNDARY CONDITIONS FOR FLOW SIMULATIONS
Five subsets of data records are required for this data set.

A. Subset 1: control parameters
(1) NCESH = No. of Cauchy boundary element sides.
(2) NCNPH = No. of Cauchy nodal points.
(3) NCPRH = No. of Cauchy-flux profiles.
(4) NCDPH = No. of data points in each of the NCPR Cauchy-flux profiles.
(5) KCAIH = Is Cauchy flux profile to be input analytically, 0 = no, 1 = yes

B. Subset 2: prescribed Cauchy-flux profiles - This sub-data-set is required only if NCESH is not 0.
NCPRH of profiles are needed.   For each profile, NCDPH of the data pair
(TQCBF(J,I),QCBF(J,I)) are needed. If these data pairs can fit in a line, we will need
NDPRH lines.
(1) TQCBF(J,I) = Time of the J-th data point in the I-th profile, (T).

(2) QCBF(J,I) = Normal Cauchy flux of the J-th data point in the I-th profile, (L**3/T/L**2); positive out from the region, negative into the region.

C. Subset 3: global node number of all compressed Cauchy nodes - At most, NCNPH records are needed for this subdata set, one each for NCNPH Cauchy nodes.
(1) NI = Compressed Cauchy node number of the first node in the sequence.
(2) NSEQ = NSEQ nodes will be generated automatically.
(3) NIAD = Increment of NI for each of the NSEQ sides.
(4) NITYP = Global node number of node NI.
(5) NTYPAD = Increment of NITYP for each of the NSEQ sides.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

D. Subset 4: type of Cauchy flux profiles assigned to each of all NCNPH sides. At most NCNPH records are needed. However, automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
(1) MI = Compressed Cauchy node number of the first node in the sequence.
(2) NSEQ = NSEQ nodes will be generated automatically.
(3) MIAD = Increment of MI for each of the NSEQ nodes.
(4) MITYP = Type of Cauchy flux profile assigned to node MI.
(5) MTYPAD = Increment of MITYP for each of the NSEQ nodes.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

E. Subset 4: Cauchy boundary element sides - Normally, NCESH records are required, one each for a Cauchy boundary element side. However, if a group of Cauchy boundary element sides appears in a regular pattern, automatic generation may be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
(1) MI = Compressed Cauchy element side number of the first element-side in a sequence.
(2) NSEQ = NSEQ subsequent Cauchy element-sides will be generated automatically.
(3) M = Global element no. to which the MI-th Cauchy side belongs.
(4) IS1 = Compressed Cauchy node number of the first node on the Cauchy element-side MI.
(5) IS2 = Compressed Cauchy node number of the second node on the Cauchy element-side MI.
(6) MIAD = Increment of MI for each of the NSEQ subsequent sides.
(7) MAD = Increment of M for each of the NSEQ subsequent Cauchy sides.
(8) IS1AD = Increment of IS1 for each of the NSEQ subsequent element-sides.
(9) IS2AD = Increment of IS2 for each of the NSEQ subsequent element-sides.

**** NOTE: A line with 9 0's is used to end this data set input.

16. NEUMANN BOUNDARY CONDITIONS FOR FLOW SIMULATIONS
    Five subsets of data records are required for this data set.

    A. Subset 1: control parameters
    (1) NNESH = No. of Neumann boundary element sides.
    (2) NNNPH = No. of Neuamnn nodal points.
    (3) NNPRH = No. of Neumann flux profiles.

(4) NNDPH = No. of data points in each of the NNPR Neumann-flux profiles.
(5) KNAIH = Is Neumann flux profile to be input analytically, 0 = no, 1 = yes

**** NOTE: A line with 5 0's is used to signal the end of this data set.

B. Subset 4: prescribed Neumann-flux profiles - This sub-data-set is required only if NNESH is not 0. NNPRH of profiles are needed. For each profile, NNDPH of the data pair (TQNBF(J,I),QNBF(J,I)) are needed. If these data pairs can fit in a line, we will need NDPRH lines.
(1) TQNBF(J,I) = Time of the J-th data point in the I-th profile, (T).
(2) QNBF(J,I) = Normal Neumann flux of the J-th data point in the I-th profile, (L**3/T/L**2); positive out from the region, negative into the region.

C. Subset 3: global node number of all compressed Neumann nodes -At most, NNNPH records are needed for this sub-data set, one each for NNNPH Neumann nodes.
(1) NI = Compressed Neumann node number of the first node in the sequence.
(2) NSEQ = NSEQ nodes will be generated automatically.
(3) NIAD = Increment of NI for each of the NSEQ sides.
(4) NITYP = Global node number of node NI.
(5) NTYPAD = Increment of NITYP for each of the NSEQ sides.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

D. Subset 4: type of Neumann flux profiles assigned to each of all NNNPH nodes. At most NNNPH records are needed. However, automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
(1) MI = Compressed Neumann node number of the first node in the sequence.
(2) NSEQ = NSEQ nodes will be generated automatically.
(3) MIAD = Increment of MI for each of the NSEQ nodes.
(4) MITYP = Type of Neumann flux profile assigned to node MI.
(5) MTYPAD = Increment of MITYP for each of the NSEQ nodes.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

E. Subset 5: Neumann boundary element sides - Normally, NNESH records are required, one each for a Neumann boundary element side. However, if a group of Neumann boundary element sides appears in a regular pattern, automatic generation may be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
(1) MI = Compressed Neumann side number of the first side in sequence.
(2) NSEQ = NSEQ subsequent Neumann sides will be generated automatically.
(3) M = Global element no. to which the MI-th Neumann side belongs.
(4) IS1 = Compressed Neumann node number of the first node on the Neumann element-side MI. (5) IS2 = Compressed Neumann node number of the second node on Neumann element-side MI.
(6) MIAD = Increment of MI for each of the NSEQ subsequent sides.
(7) MAD = Increment of M for each of the NSEQ subsequent Neumann side.
(8) IS1AD = Increment of IS1 for each of the NSEQ subsequent Neumann side.
(9) IS2AD = Increment of IS2 for each of the NSEQ subsequent Neumann side.
**** NOTE: A line with 9 0's is used to end this data set input.

17. BASIC REAL AND INTEGER PARAMETERS FOR TRANSPORT
    Four subsets of free-formatted data records are required for this data set.

    A. Subset 1: This group of data reads the integer control parameter for transport. Five variables are
            included.
    (1) NITERT = No. of iterations allowed for solving nonlinear equations.
    (2) NPITERT = No. of iterations allowed for pointwise iteration.
    (3) KSTRT = Auxiliary storage output control: 0 = No storage, 1 = Output stored on logical unit 11.
    (4) KSST = Steady state control for flow: 0 = Steady state solution is desired,
            1 = Only transient solutions are desired.
    (5) KVIT = Velocity input control:
            -1 = Card input for velocity and moisture content.
            1 = Steady-state velocity and moisture content input via logical unit 12.
            2 = transient velocity and moisture content input via auxiliary storage logical unit 12.
    (6) MICONF = Index of the simulation of microbial configuation
            0 = mobile microbes
            1 = immobile microbes

    B. Subset 2: This group of data has seven variables.
    (1) TOLAT = Steady-state convergence criteria.
    (2) TOLBT = Transient-state convergence criteria.
    (3) WT = Time derivative weighting factor:
            0.5 = Crank-Nicolson central,
            1.0 = Backward difference and/or mid-difference.
    (4) WV = Time integration factor for the velocity terms:
            0.0 = Forward difference (explicit),
            0.5 = Central difference,
            1.0 = Backward difference (implicit).
    (5) OMET = Iteration parameter for solving the nonlinear matrix equation:
            $0.0 \sim 1.0$ = Under-relaxation,
            1.0 = Exact relaxation,
            $1.0 \sim 2.0$ = Over-relaxation.
    (6) OMIT = Relaxation parameter for solving the matrix equation pointwisely:
            $0.0 \sim 1.0$ = Under-relaxation,
            1.0 = Exact relaxation,
            $1.0 \sim 2.0$ = Over-relaxation.
    (7) APHAG = Upstream weighting factor used if IOPTIM.EQ.0.
            NOTE: (1) The value is between 0.0 and 1.5,
                    (2) If APHAG.GE.1.34, the program will choose appropriate values of weighting
                        factor.

    C. Subset 3: It contains the following 10 variables (FREE FORMAT)
    (1) IZOOM = Is zooming needed for advection computation? 0 = No, 1 = Yes.

(2) IDZOOM = Is zooming needed for dispersion computation? 0 = No, 1 = Yes.

(3) IEPC = Is EPCOF scheme included? 0 = No, 1 = Yes.

(4) NXA = No. of regularly refined grids for the advection step in the X-drection in an element.

(5) NZA = No. of regularly refined grids for the advection step in the Z-drection in an element.

(6) NXD = No. of dispersion fine grids in the X-direction.

(7) NZD = No. of dispersion fine grids in the Z-direction.

(8) NXW = No. of sub-grid points for the sub-element tracking in the X-direction in an element.

(9) NZW = No. of sub-grid points for the sub-element tracking in the Z-direction in an element.

(10) IDETQ = Index of particle tracking patten;
       1 = Average velocity is used (more accurate);
       2 = Single velocity of the starting point is used (less computation).

D. Subset 4: It reads the following 2 variables (FREE FORMAT)

(1) ADPEPS = Error tolerance of relative concentration and nonlinear convergence criteria.

(2) ADPARM = Error tolerance of concentration relative to maximum concentration.

18. INITIAL/PRE-INITIAL CONDITIONS FOR TRANSPORT
There are seven subsets in this data set:

A. Subset 1: This group of data reads the initial conditions (M/L^3) of transport for Microbe #1. There are 6 variables in each line of this subset (FREE FORMAT)

(1) NI = The global node number of the 1-st node of the line.

(2) NSEQ = No. of subsequent nodes to be counted in the line.

(3) NAD = The increment of global node no. for each subsequent node in the line.

(4) FNI = The initial/pre-initial total concentration of the 1-st node of the line.

(5) FAD = The increment of total concentration for each subsequent node in the line.

(6) FRD = The increment ratio of total concentration for each subsequent node in the line. (i.e., the increment increases with rate 1.0+FRD). Usually, it is set to be 0 for transport.

NOTE: All the values of variables in the last line of the subset should be 0.

B. Subset 2 ~ Subset NCC: The 2-nd to NCC-th subsets reads the initial conditions of transport for Microbe #2, Microbe #3, substrate, Oxygen, Nitrate, nutrient, and up to the NCC-th component.

19. MICROBE-CHEMICAL INTERACTION CONSTANTS
Eleven subsets of FREE-FORMATTED data are needed.

A. Subset 1: Four parameters describing the specific growth rate of microbes (1/T) are needed in this record. If there are no microbes in this system, the following four numbers have to be set to zeros.

(1) GRATE(1) = Maximum specific oxygen-based growth rate for microbe #1. ($\mu_o^{(1)}$ in Eqs. (2.9) ~ (2.15)).

(2) GRATE(2) = Maximum specific nitrate-based growth rate for microbe #2. ($\mu_n^{(2)}$ in Eqs. (2.9) ~ (2.15)).

(3) GRATE(3) = Maximum specific oxygen-based growth rate for microbe #3. ($\mu_o^{(3)}$ in Eqs. (2.9) ~

$$(2.15)).$$

(4) GRATE(4) = Maximum specific nitrate-based growth rate for microbe #3. ($\mu_n^{(3)}$ in Eqs. (2.9) ~ (2.15)).

B. Subset 2 : Four yield coefficient (M/M) are needed in this record and these four values can not be zeros.

(1) YCOEFF(1) = Yield coefficient for microbe #1 utilizing Oxygen. ($Y_o^{(1)}$ in Eqs. (2.9) & (2.12)).

(2) YCOEFF(2) = Yield coefficient for microbe #2 utilizing Nitrate. ($Y_n^{(2)}$ in Eqs. (2.9) & (2.12)).

(3) YCOEFF(3) = Yield coefficient for microbe #3 utilizing Oxygen. ($Y_o^{(3)}$ in Eqs. (2.9) & (2.12)).

(4) YCOEFF(4) = Yield coefficient for microbe #3 utilizing Nitrate. ($Y_n^{(3)}$ in Eqs. (2.9) & (2.12)).

C. Subset 3: Four retarded substrate saturation constants (M/L^3) are needed in this record.

(1) RTARDS(1) = Retarded substrate saturation constant under aerobic conditions with respect to microbe #1. ($K_{so}^{(1)}$ in Eqs. (2.9) ~ (2.15)).

(2) RTARDS(2) = Retarded substrate saturation constant under anaerobic conditions with respect to microbe #2. ($K_{sn}^{(2)}$ in Eqs. (2.9) ~ (2.15)).

(3) RTARDS(3) = Retarded substrate saturation constant under aerobic conditions with respect to microbe #3. ($K_{so}^{(3)}$ in Eqs. (2.9) ~ (2.15)).

(4) RTARDS(4) = Retarded substrate saturation constant under anaerobic conditions with respect to microbe #3. ($K_{sn}^{(3)}$ in Eqs. (2.9) ~ (2.15)).

D. Subset 4: Four retarded Oxygen or Nitrate saturation constant (M/L^3) are needed in this record.

(1) RTARDO(1) = Retarded Oxygen saturation constant under aerobic conditions with respect to microbe #1. ($K_o^{(1)}$ in Eqs. (2.9) ~ (2.15)).

(2) RTARDO(2) = Retarded Nitrate saturation constant under anaerobic conditions with respect to microbe #2. ($K_n^{(2)}$ in Eqs. (2.9) ~ (2.15)).

(3) RTARDO(3) = Retarded Oxygen saturation constant under aerobic conditions with respect to microbe #3. ($K_o^{(3)}$ in Eqs. (2.9) ~ (2.15)).

(4) RTARDO(4) = Retarded Nitrate saturation constant under anaerobic conditions with respect to microbe #3. ($K_n^{(3)}$ in Eqs. (2.9) ~ (2.15)).

E. Subset 5: Four retarded nutrient saturation constant (M/L^3) are needed in this record.

(1) RTARDN(1) = Retarded nutrient saturation constant under aerobic conditions with respect to microbe #1. ($K_{po}^{(1)}$ in Eqs. (2.9) ~ (2.15)).

(2) RTARDN(2) = Retarded nutrient saturation constant under anaerobic conditions with respect to microbe #2. ($K_{pn}^{(2)}$ in Eqs. (2.9) ~ (2.15)).

(3) RTARDN(3) = Retarded nutrient saturation constant under aerobic conditions with respect to microbe #3. ($K_{po}^{(3)}$ in Eqs. (2.9) ~ (2.15)).

(4) RTARDN(4) = Retarded nutrient saturation constant under anaerobic conditions with respect to microbe #3. ($K_{pn}^{(3)}$ in Eqs. (2.9) ~ (2.15)).

F. Subset 6: Four Oxygen-use of Nitrate-use coefficient for synthesis are needed in this record.

(1) SCOEFF(1) = Oxygen-use coefficient for synthesis by microbe #1. ($\gamma_o^{(1)}$ in Eq. (2.10)).

(2) SCOEFF(2) = Nitrate-use coefficient for synthesis by microbe #2. ($\gamma_n^{(2)}$ in Eq. (2.11)).

(3) SCOEFF(3) = Oxygen-use coefficient for synthesis by microbe #3. ($\gamma_o^{(3)}$ in Eq. (2.10)).

(4) SCOEFF(4) = Nitrate-use coefficient for synthesis by microbe #3. ($\gamma_n^{(3)}$ in Eq. (2.11)).

G. Subset 7: Four Oxygen-use of Nitrate-use coefficient for energy are needed in this record.

(1) ECOEFF(1) = Oxygen-use coefficient for energy by microbe #1. ($\alpha_o^{(1)}$ in Eq. (2.10)).
(2) ECOEFF(2) = Nitrate-use coefficient for energy by microbe #2. ($\alpha_n^{(2)}$ in Eq. (2.11)).
(3) ECOEFF(3) = Oxygen-use coefficient for energy by microbe #3. ($\alpha_o^{(3)}$ in Eq. (2.10)).
(4) ECOEFF(4) = Nitrate-use coefficient for energy by microbe #3. ($\alpha_n^{(3)}$ in Eq. (2.11)).

H. Subset 8: Four microbial decay coefficient (1/T) are needed in this record.
(1) DCOEFF(1) = Microbial decay coefficient of aerobic respiration of microbe #1. ($\lambda_o^{(1)}$ in Eqs. (2.13) & (2.15)).
(2) DCOEFF(2) = Microbial decay coefficient of anaerobic respiration of microbe #2. ($\lambda_n^{(2)}$ in Eqs. (2.14) & (2.15)).
(3) DCOEFF(3) = Microbial decay coefficient of aerobic respiration of microbe #3. ($\lambda_o^{(3)}$ in Eqs. (2.13) & (2.15)).
(4) DCOEFF(4) = Microbial decay coefficient of anaerobic respiration of microbe #3. ($\lambda_n^{(3)}$ in Eqs. (2.14) & (2.15)).

I. Subset 9: Four Oxygen or Nitrate saturation constants (M/L^3) for decay are needed in this record.
(1) SATURC(1) = Oxygen-saturation constant for decay with respect to microbe #1. ($\Gamma_o^{(1)}$ in Eq. (2.10)).
(2) SATURC(2) = Nitrate-saturation constant for decay with respect to microbe #2. ($\Gamma_n^{(2)}$ in Eq. (2.11)).
(3) SATURC(3) = Oxygen-saturation constant for decay with respect to microbe #3. ($\Gamma_o^{(3)}$ in Eq. (2.10)).
(4) SATURC(4) = Nitrate-saturation constant for decay with respect to microbe #3. ($\Gamma_n^{(3)}$ in Eq. (2.11)).

J. Subset 10: Four nutrient-use coefficient for the production are needed in this record.
(1) PCOEFF(1) = Nutrient-use coefficient for the production of microbe #1 with aerobic respiration. ($\epsilon_o^{(1)}$ in Eq. (2.12)).
(2) PCOEFF(2) = Nutrient-use coefficient for the production of microbe #2 with anaerobic respiration. ($\epsilon_n^{(2)}$ in Eq. (2.12)).
(3) PCOEFF(3) = Nutrient-use coefficient for the production of microbe #3 with aerobic respiration. ($\epsilon_o^{(3)}$ in Eq. (2.12)).
(4) PCOEFF(4) = Nutrient-use coefficient for the production of microbe #3 with anaerobic respiration. ($\epsilon_n^{(3)}$ in Eq. (2.12)).

K. Subset 11: One variable (M/L^3) is included in this record.
(1) COFK = Inhibition coefficient.

20. ELEMENT (DISTRIBUTED) SOURCE/SINK FOR TRANSPORT SIMULATIONS
   Ten subsets of free-formatted data records are required in this data set.

   A. Subset 1: control parameters
   (1) NSEL = No. of source/sink elements.
   (2) NSPR = No. of source profiles, should be .GE. 1.
   (3) NSDP = No. of data points in each profile, should be .GE. 2.
   (4) KSAI = Is element-source/sink profile to be input analytically? 0 = no, 1 = yes.

B. Subset 2: source/sink profile - This sub-data-set is needed if and only if NSEL .GT. 0. For each sub-data-record, NSDP of the data group (TSOSF(J,I),SOSF(J,I,1),SOSF(J,I,2)) are required. If this sub-data-record can be fitted in a line, we will need NSPR lines.

(1) TSOSF(J,I) = Time of J-th data point in I-th profile, (T).

(2). SOSF(J,I,1) = Source/sink flow rate of the J-th data point in the I-th profile, (L**3/T/L**3); positive for source and negative for sink.

(3) SOSF(J,I,2) = Source/sink concentration of the J-th data point in the I-th profile, (M/L**3).

C. Subset 3: global source/sink element number. NSEL data points are required for this record.

(1) LES(I) = Global element number of the I-th compressed distributed source/sink element.

D. Subsets 4: Source type assigned to each element for microbe #1 - Usually one record per element. However, automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following.

(1) MI = Global element number of the first element in the sequence.

(2) NSEQ = NSEQ elements will be generated automatically.

(3) MAD = Increment of element number for each of the NSEQ elements.

(4) MITYP = Source type in element MI.

(5) MTYPAD = Increment of MITYP for each of the NSEQ elements.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

E. Subset 5 ~ Subset NCC+3: Source type assigned to each element for microbe #2, microbe #3, substrate, Oxygen, Nitrate, nutrient, and up to the NCC-th component. The input format is the same as subset 4.

21. POINT (WELL) SOURCE/SINK DATA FOR TRANSPORT SIMULATION

Ten subsets of data records are required for this data set.

A. Subset 1: control parameters

(1) NWNP = No. of well or point source/sink nodes.

(2) NWPR = No. of well or point source/sink strength profiles.

(3) NWDP = No. of data points in each of the NWPR profiles.

(4) KWAI = Is well-source/sink profile to be input analytically? 0 = no, 1 = yes.

B. Subset 2: source/sink profiles - This group of data is needed if and only if NWNP .GT. 0. For each sub-data-record, NWDP of the data group (TWSSF(J,I), WSSF(J,I,1), WSSF(J,I,2)) are required. If this sub-data-record can be fitted in a line, we will need NWPR lines.

(1) TWSSF(J,I) = Time of J-th data point in I-th profile, (T).

(2) WSSF(J,I,1) = Source/sink flow rate of the J-th data point in the I-th profile, (L**3/T/L**3); positive for source and negative for sink.

(3) WSSF(J,I,2) = Source/sink concentration of the J-th data point in the I-th profile, (M/L**3).

C. Subset 3: global source/sink node number - This group of data is needed if and only if NWNP .GT. 0. NWNP data points are required for this record.

(1) NPW(I) = Global node number of the I-th compressed point source/sink node.

D. Subset 4: Source type assigned to each well for microbe #1 - Usually one record per element. However,automatic generation can be made.
(1) NI = Compressed point source/sink node number of the first node in a sequence.
(2) NSEQ = NSEQ nodes will contain the source types that will be automatically generated.
(3) NIAD = Increment of NI for each of the NSEQ nodes.
(4) NITYP = Source type in node NI.
(5) NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's must be used to signal the end of this data set.

E. Subset 5 ~ Subset NCC+3: Source type assigned to each well for microbe #2, microbe #3, substrate, Oxygen, Nitrate, nutrient, and up to the NCC-th component.

22. RUN-IN/FLOW-OUT (VARIABLE) BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS

Ten subsets of data records are required for this data set.

A. Subset 1: control parameters
(1) NVES = No. of variable boundary element sides
(2) NVNP = No. of variable boundary nodal points
(3) NRPR = No. of incoming fluid concentration profiles to be applied to variable boundary element sides.
(4) NRDP = No. of data points in each of the NRPR profiles.
(5) KRAI = Is incoming concentration profile to be input analytically? 0 = no, 1 = yes.

B. Subset 2: variable boundary flux profile - NRPR records are needed. Each record contains NRDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and run-in flow-out concentrations, respectively as follows:
(1) TCRSF(J,I) = Time of the J-th data point on the I-th run-in concentration profile, (T),
(2) CRSF(J,I) = Concentration of the J-th data point on the I-th profile, (M/L**3).

C. Subset 3: Run-in concentration type assigned to each of all NVES sides for microbe #1. Usually one record per variable element side. However, automatic generation can be made. Each record contains 5 variable and is FREE-FORMATTED.
(1) MI = Compressed VB element side of the first side in a sequence.
(2) NSEQ = NSEQ subsequent sides will be generated automatically.
(3) MIAD = Increment of MI for each of NSEQ subsequent sides.
(4) MITYP = Type of concentration profile assigned to side MI.
(5) MTYPAD = Increment of MITYP for each of the NSEQ subsequent sides.

**** NOTE: A record with 5 0's must be used to signal the end of this subdata set.

D. Subset 4 ~ Subsets 4 through NCC+2: Run-in concentration type assigned to each of all NVES sides for microbe #2, microbe #3, substrate, Oxygen, Nitrate, nutrient, and up to the NCC-th component. The input format is the same as subset 3.

E. Subset NCC+3: global nodal number of all run-in flow-out boundary nodes Usually NVNP records

are needed for this sub-data set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
(1) NI = Compressed VB node number of the first node in a sequence.
(2) NSEQ = NSEQ subsequent nodes will be generated automatically.
(3) NIAD = Increment for NI for each of the NSEQ nodes.
(4) NODE = Global nodal number of the node NI.
(5) NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's is used to signal end of this subdata set.

F. Subset NCC+4: Specification of run-in boundary element sides - Normally, NVES records are required, one each for a VB element side. However, if a group of VB element sides appears in a regular pattern, automatic generation may be made. Each record contains 11 variables and is FREE-FORMATTED.
(1) MI = Compressed VB element side number of the first side in a sequence.
(2) NSEQ = NSEQ subsequent VB element sides will be generated automatically.
(3) M = Global element no. to which the MI-th VB side belongs.
(4) IS1 = Global node number of the first node of element side MI.
(5) IS2 = Global node number of the second node of element side MI.
(6) MIAD = Increment of MI for each of the NSEQ subsequent VB element sides.
(7) MAD = Increment of M for each of the NSEQ subsequent VB element sides.
(8) IS1AD = Increment of IS1 for each of the NSEQ subsequent element sides.
(9) IS2AD = Increment of IS2 for each of the NSEQ subsequent element sides.

**** NOTE: A record with 9 0's is used to signal the end of this subdata set.

23. DIRICHLET BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS

Five subsets of data records are required for this data set.

A. Subset 1: control parameters
(1) NDNP = No. of Dirichlet nodes, should be .GE. 1.
(2) NDPR = No. of Dirichlet profiles, should be .GE. 1.
(3) NDDP = No. of data points in each profile, should be .GE. 2.
(4) KDAI = Is Dirichlet boundary value profile to be input analytically? 0 = no, 1 = yes.

B. Subset 2: Dirichlet-concentration profiles - NDPR records are needed. Each record contains NDDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Dirichlet concentrations, respectively as follows:
(1) TCDBF(J,I) = Time of J-th data point in I-th Dirichlet-concentration profile, (T).
(2) CDBF(J,I) = Concentration of J-th data point in I-th Dirichlet-concentration profile, (M/L**3).

C. Subset 3: Dirichlet concentration types assigned to Dirichlet nodes for Microbe # 1 - Normally one record per Dirichlet node, i.e., a total of NDNP records, is needed. However, if the Dirichlet nodes appear in a regular pattern, automatic generation may be made. Each record contains 5 variables and is FREE-FORMATTED.
(1) NI = Compressed Dirichlet node number of the first node in the sequence.

125

(2) NSEQ = NSEQ nodes will contain the Dirichlet concentration types that will be automatically generated.
(3) NIAD = Increment of NI for each of the NSEQ nodes.
(4) NITYP = Dirichlet concentration type in node NI.
(5) NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's must be used to signal the end of this data set.

D. Subset 4 ~ Subset NCC+2: Dirichlet concentration type assigned to each of all NDNP points for microbe #2, microbe #3, substrate, Oxygen, Nitrate, nutrient, and up to the NCC-th component. The input format is the same as subset .

E. Subset NCC+3: global node number of compressed Dirichlet nodes - One record is needed for this sub-data set, which contains NDNP variables and is FREE-FORMATTED.
(1) NPDB(I) = Global node number of the I-th compressed Dirichlet node.

24. CAUCHY BOUNDARY CONDITIONS FOR TRANSPORT SIMULATION

Six subsets of data records are required for this data set.

A. Subset 1: control parameters
(1) NCES = No. of Cauchy element sides.
(2) NCNP = No. of Cauchy nodal points.
(3) NCPR = No. of Cauchy-flux profiles.
(4) NCDP = No. of data points on each Cauchy-flux profile.
(5) KCAI = Is Cauchy flux profile to be input analytically? 0 = no, 1 = yes.

B. Subset 2: Cauchy flux profiles - NCPR records are needed. Each record contains NCDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Cauchy flux, respectively, as follows:
(1) TQCBF(J,I) = Time of the J-th data point in the I-th Cauchy flux profile, (T).
(2) QCBF(J,I) = Value of Cauchy flux of the J-th data point in the I-th Cauchy flux profile, (M/T/L**2).

C. Subset 3: Cauchy flux type assigned to each of all NCNP nodes -Usually one record per Cauchy node. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
(1) MI = Compressed Cauchy boundary node number of the first node in a sequence.
(2) NSEQ = NSEQ subsequent nodes will be generated automatically.
(3) MIAD = Increment of MI for each of NSEQ subsequent nodes.
(4) MITYP = Type of Cauchy flux profile assigned to node MI.
(5) MTYPAD = Increment of MITYP for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's must be used to signal the end of this subdata set.

D. Subset 4 ~ Subset NCC+2: Cauchy flux type assigned to each of all NCNP nodes for microbe #2,

microbe #3, substrate, Oxygen, Nitrate, nutrient, and up to the NCC-th component. The input format is the same as subset 3.

E. Subset NCC+3: global nodal number of all Cauchy boundary nodes -Usually NCNP records are needed for this sub-data set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
(1) NI = Compressed Cauchy boundary node number of the first node in a sequence.
(2) NSEQ = NSEQ subsequent nodes will be generated automatically.
(3) NIAD = Increment for NI for each of the NSEQ nodes.
(4) NODE = Global nodal number of the node NI.
(5) NODEAD = Increment of the global nodal number for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's is used to signal end of this subdata set.

F. Subset NCC+4: specification of Cauchy boundary element sides -Normally, NCES records are required, one each for a Cauchy boundary element side. However, if a group of Cauchy element sides appears in a regular pattern, automatic generation may be made. Each record contains 9 variables and is FREE-FORMATTED.
(1) MI = Compressed Cauchy boundary element side number of the first element side in a sequence.
(2) NSEQ = NSEQ subsequent Cauchy boundary element sides will be generated automatically.
(3) M = Global element number to which the MI-th Cauchy element side belongs.
(4) IS1 = Compressed Cauchy node number of the first node of element side MI.
(5) IS2 = Compressed Cauchy node number of the second node of element side MI.
(6) MIAD = Increment of MI for each of the NSEQ subsequent Cauchy boundary element sides.
(7) MAD = Increment of M for each of the NSEQ subsequent Cauchy boundary element sides.
(8) IS1AD = Increment of IS1 for each of the NSEQ subsequent element sides.
(9) IS2AD = Increment of IS2 for each of the NSEQ subsequent element sides.

**** NOTE: A record with 9 0's is used to signal the end of this subdata set.

25. NEUMANN BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS

Six subsets of data records are required for this data set.

A. Subset 1: control parameters
(1) NNES = No. of Neumann element sides.
(2) NNNP = No. of Neumann nodal points.
(3) NNPR = No. of Neumann-flux profiles.
(4) NNDP = No. of data points on each Neumann-flux profile.
(5) KSAI = Is Neumann flux profile to be input analytically? 0 = no, 1 = yes.

B. Subset 2: Neumann flux profiles - NNPR records are needed. Each record contains NNDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Neumann flux, respectively, as follows:
(1) TQNBF(J,I) = Time of the J-th data point in the I-th Neumann flux profile, (T).
(2) QNBF(J,I) = Value of Neumann flux of the J-th data point in the I-th Neumann flux profile, (M/T/L**2).

C. Subset 3: Neumann flux type assigned to each of all NNNP nodes -Usually one record per Neumann boundary node. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
(1) MI = Compressed Neumann boundary node nmber of the first node in a sequence.
(2) NSEQ = NSEQ subsequent nodes will be generated automatically.
(3) MIAD = Increment of MI for each of NSEQ subsequent nodes.
(4) MITYP = Type of Neumann flux profile assigned to node MI.
(5) MTYPAD = Increment of MITYP for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's must be used to signal the end of this sub-data set.

D. Subset 4 ~ Subset NCC+2: Neumann flux type assigned to each of all NNNP nodes for microbe #2, microbe #3, substrate, Oxygen, Nitrate, nutrient, and up to the NCC-th component. The input format is the same as subset 3.

E. Subset NCC+3: global nodal number of all Neumann boundary nodes -Usually NNNP records are needed for this sub-data set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
(1) NI = Compressed Neumann boundary node number of the first node in a sequence.
(2) NSEQ = NSEQ subsequent nodes will be generated automatically.
(3) NIAD = Increment for NI for each of the NSEQ nodes.
(4) NODE = Global nodal number of the node NI.
(5) NODEAD = Increment of the global nodal number for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's is used to signal end of this subdata set.

F. Subset NCC+4: specification of Neumann boundary element sides -Normally, NNES records are required, one each for a Neumann boundary element side. However, if a group of Neumann element sides appears in a regular pattern, automatic generation may be made. Each record contains 9 variable and is FREE-FORMATTED.
(1) MI = Compressed Neumann boundary element side number of the first element side in a sequence.
(2) NSEQ = NSEQ subsequent Neumann boundary element sides will be generated automatically.
(3) M = Global element no. to which the MI-th Neumann boundary side.
(4) IS1 = Compressed Neumann node number of the first node of element side MI.
(5) IS2 = Compressed Neumann node number of the second node of element side MI.
(6) MIAD = Increment of MI for each of the NSEQ subsequent ides.
(7) MAD = Increment of M for each of the NSEQ subsequent Neumann boundary sides.
(8) IS1AD = Increment of IS1 for each of the NSEQ subsequent element sides.
(9) IS2AD = Increment of IS2 for each of the NSEQ subsequent element sides.

**** NOTE: A record with 9 0's is used to signal the end of this subdata set.

26. HYDROLOGICAL VARIABLES

This data set is needed if and only if KVIT .LE. 0. When KVIT .LE. 0, three sub-data sets are needed, two groups for the velocity field and the other group for the moisture content.

128

A. Subset 1: velocity field (x-direction) - Usually NNP records are needed. However, if velocity appears in a regular pattern, automatic generation can be made. Each record contains 6 variables and is FREE-FORMATTED.
(1) NI = Node number of the first node in a sequence
(2) NSEQ = NSEQ subsequent nodes will be automatically generated,
(3) NIAD = Increment of node number in each of the NSEQ subsequent nodes,
(4) VXNI = x-velocity component at node NI, (L/T),
(5) VXAD = Increment of VXNI for each of the NSEQ subsequent nodes, (L/T),
(6)   0.0

B. Subset 2: velocity field (z-direction) - Usually NNP records are needed. However, if velocity appears in a regular pattern, automatic generation can be made. Each record contains 6 variables and is FREE-FORMATTED.
(1) NI = Node number of the first node in a sequence
(2) NSEQ = NSEQ subsequent nodes will be automatically generated,
(3) NIAD = Increment of node number in each of the NSEQ subsequent nodes,
(4) VZNI = z-velocity component at node NI, (L/T),
(5) VZAD = Increment of VZNI for each of the NSEQ subsequent nodes, (L/T),
(6)   0.0

**** NOTE: A record with 6 0's is used to signal the end of this sub-data set.

C. Subset 3: moisture content field - Usually, NEL records are needed. However, if moisture content appears in a regular pattern, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
(1) MI = Element number of the first element in a sequence,
(2) NSEQ = NSEQ subsequent elements will be automatically generated,
(3) MIAD = Increment of MI for each of NSEQ subsequent elements,
(4) THNI = Moisture content of element NI, (Decimal point),
(5) THNIAD = Increment of THNI for NSEQ subsequent elements, (Decimal point).
(6) 0.0

**** NOTE: A record with 6 0's is used to signal the end of this subdata set.

27. END OF JOB

If another problem is to be run, then input begins again with input data set 1. If termination of the job is desired, a blank card must be inserted at the end of the data set.

REFERENCES

Bachelor, G. A., D. E. Cawlfield, F. T. Lindstrom, and L. Boersma, Denitrification in nonhomogeneous laboratory scale aquifers: 5: User's manual for the mathematical model LT3VSI, A draft report, 1990.

Benfield, L. D., and F. J. Molz, A model for the activated sludge process which considers wastewater characteristics, flux behavior, and microbial population, *Biotechnol. Bioeng.*, 26, 352-361, 1984.

Cheng, J. R., H. P. Cheng, and G. T. Yeh. A Lagrangian-Eulerian Method with Adaptively Local Zooming and Peak/Valley Capturing Approach to Solve Two-Dimensional Advection-Diffusion Transport Equations. International J. of Numerical Methods in Engineering. 1995 (in press).

Freeze, R. A., Role of subsurface flow in generating surface runoff: 1. Base flow contribution to channel flow, *Water Resour. Res.*, 8, 609-623, 1972a.

Freeze, R. A., Role of subsurface flow in generating surface runoff: 2. Upstream source areas, *Water Resour. Res.*, 8, 1272-1283, 1972b.

Herbert, D., Some principles of continuous culture, in *Recent Progress in Microbiology*, edited by G. Tunevall, Blackwell Scientific Publishers, Oxford, England, 1958.

Huyakorn, P. S., E. P. Springer, V. Guvanasen, and T. D. Wadsworth, A three-dimensional finite-element model for simulating water flow in variably saturated porous media, *Water Resources Research*, Vol. 22, No. 13, 1790-1808, 1986.

MacQuarrie, K. T. B. and E. A. Sudicky, Simulation of biodegradable organic contaminants in groundwater, 2. plume behavior in uniform and random flow fields, *Water Resour. Res.,* 26(2), 223-239, 1990.

Molz, F. J. M. A. Widdowson, and L. D. Benfield, Simulation of microbial growth dynamics coupled to nutrient and oxygen transport in porous media, *Water Resour. Res.,* 22(8), 1207-1216, 1986.

van Genuchten, M. Th., A closed form equation for predicting the hydraulic conductivity of unsaturated soils, *Soil Science Society of America Journal*, 44, 892-898, 1980.

Widdowson, M. A., F. J. Molz, and L. D. Benfield, A numerical transport model for oxygen- and nitrate-based respiration linked to substrate and nutrient availability in porous media, *Water Resour. Res.*, 24(9), 1553-1565, 1988.

Yeh, G. T., and D. S. Ward, FEMWATER: A finite-element model of water flow through saturated-unsaturated porous media, Rep. ORNL-5567, Oak Ridge Nat. Lab., Oak Ridge, Tenn., 37831, 137 pp., 1980.

Yeh, G. T., and D. S. Ward, FEMWASTE: A finite-element model of waste transport through saturated-unsaturated porous media, Rep. ORNL-5601, Oak Ridge Nat. Lab., Oak Ridge, Tenn., 37831, 137 pp., 1981.

Yeh, G. T., FEMWATER: A finite element model of water flow through saturated-unsaturated porous media, First Revision, Rep. ORNL-5567/R1, Oak Ridge Nat. Lab., Oak Ridge, Tenn., 37831, 258 pp., 1987.

Yeh, G. T., A Lagrangian-Eulerian method with zoomable hidden fine mesh approach to solving advection-dispersion equations, *Water Resources Research* Vol. 26, No. 6, 1133-1144, 1990.

Yeh, G. T., J. R. Chang, and T. E. Short, An exact peak capturing and oscillation-free scheme to solve advection-dispersion transport equations, *Water Resour. Res.*, 28(11), 2937-2951, 1992.

Yeh, G. T., Class notes: CE597C: Computational Subsurface Hydrology Part II, The Pennsylvania State University, University Park, PA., 16802, Spring semester 1992.

Yeh, G. T., 3DFEMWATER: A three-dimensional finite element model of WATER flow through saturated-unsaturated media: Version 2.0, Short course notes of Simulation of Subsurface Flow and Contaminant Transport by Finite Element and Analytical Methods, The Pennsylvania State University, University Park, PA., 16802, May 1993.

Yeh, G. T., 3DLEWASTE: A three-dimensional hybrid Lagrangian-Eulerian finite element model of WASTE transport through saturated-unsaturated media: Version 2.0, Short course notes of Simulation of Subsurface Flow and Contaminant Transport by Finite Element and Analytical Methods, The Pennsylvania State University, University Park, PA., 16802, May 1993.

Yeh, G. T., J. R. Chang, J. P. Gwo, H. C. Lin, D. R. Richards, and W. D. Martin, 3DSALT: A three-dimensional finite element model of density-dependent flow and transport through saturate-unsaturated media, Instruction Report HL-94-1, US Army Corps of Engineers, 1994.